This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

Technical Report of the ISIS LLRF

ESS-Bilbao RF group,

UPV Control Department

February 2010

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

Authors	e-mail address
Hooman Hassanzadegan	hhassan@essbilbao.com
Nagore Garmendia	ngarmendia@essbilbao.com
Supervisor	e-mail address
Víctor Etxebarría	victor@we.lc.ehu.es

Index

1. ISIS RFQ Specifications 5
2. Design description of the ISIS LLRF 6
   - 2.1. Amplitude and phase loops 6
   - 2.2. Tuning loop 7
3. Implementation of the ISIS LLRF 8
   - 3.1. LLRF rack 8
   - 3.2. Power Supply Unit 9
   - 3.3. RF Distribution Unit 10
   - 3.4. Analog Front-End Unit 12
   - 3.5. Tuning Unit 14
   - 3.6. Tuner Driver Unit 16
   - 3.7. Digital Unit 18
   - 3.8. PC Unit 19
   - 3.9. Mock-up Cavity 20
4. LLRF unit schematics 20
   - 4.1. RF Distribution Unit 21
   - 4.2. Analog Front-End Unit 22
   - 4.3. Tuning Unit 23
   - 4.4. Tuner Driver Unit 24
5. Unit interconnections 25
6. Implementation of the amplitude and phase regulation loops 30
7. Implementation of the tuning loop 32
8. LLRF GUI (Graphical User Interface) 33
   - 8.1. LLRF parameters 36
9. Setting the LLRF Parameters 40
   - 9.1. Amplitude and phase loops: 40
   - 9.2. Tuning loop: 42
10. FPGA programming procedure 43
    - 10.1. Definition of the FPGA algorithm in MATLAB-Simulink 43
    - 10.2. Compilation 45
    - 10.3. Inserting the cosim block into a cosim model 47
    - 10.4. Programming the FPGA 48

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

Index of Tables

Table 1: Specifications of the ISIS RFQ system 5
Table 2: RF Distribution Unit interconnections 11
Table 3: Power budget estimation of the RF Distribution Unit 11
Table 4: Input and output ports of the Analog Front–end Unit 13
Table 5: Tuning Unit ports and their functions 16
Table 6: Input/output ports of the Tuner Driver Unit and their functions 18
Table 7: Resonant cavity measurements 20
Table 8: Power Supply Unit Interconnections 25
Table 9: RF Distribution Unit Interconnections 26
Table 10: Analog Front End Unit Interconnections 27
Table 11: Tuning Unit Interconnections 28
Table 12: Tuner Driver Unit Interconnections 29
Table 13: Amplitude and phase loop parameters 36
Table 14: Tuning loop parameters 38
Table 15: Amplitude, phase and tuning loop parameters 39

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

Abstract:

This report details the design and implementation of the LLRF system for the RFQ of the ISIS Front End Stand. The LLRF system includes three feedback loops to regulate the amplitude and phase of the RFQ field as well as the RFQ resonance frequency. The design is based on a fast analog front-end for RF-baseband conversion and a model-based Virtex-4 FPGA unit for signal processing and PI regulation. Complexity of the LLRF timing is significantly reduced in the current design and the LLRF requirements are fulfilled by utilizing the RF-baseband conversion method compared to the RF-IF approach. A GUI has also been implemented in MATLAB-Simulink so that the user can set the control parameters and monitor the read-back values while the LLRF system is being tested. The report begins by presenting the main parameters of the ISIS RFQ and continues by presenting the detailed design of the LLRF units as well as the Model Based FPGA program in MATLAB-Simulink.

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

1. ISIS RFQ Specifications

Table 1 summarizes the specifications of the ISIS pulsed RFQ system:

Table 1: Specifications of the ISIS RFQ system

Nominal frequency	324	MHz
No. of LLRF systems	1	
Amplitude stability	1	%
Phase stability	1	º
Tuning range	1	MHz
Unloaded Q	9000	
RF power (peak)	1000	kW
RF voltage	3000	KV
Synchronous phase	0	º
RF pulse width (min)	250	µs
RF pulse width (max)	2	ms
Pulse repetition rate	50	Hz
Ratio of the Copper to beam power	5 to 1 (app.)	

The width of the beam pulse will be shorter than the one of the RF pulse. The beam and RF pulses are synchronized so that the beam pulse arrives right after the amplitude and phase of the RF field in the RFQ have settled. The settling time should be less than 100 µs.

The magnitude of the cavity input power can exceed the nominal one (the higher limit is the maximum output power of the Klystron) during the settling time to improve the time needed to reach the nominal voltage in the cavity. Also the loop can be designed so that the cavity voltage undergoes some oscillations before settling to improve this rise time. The LLRF design engineer can play with these parameters to have the best performance.

The required amplitude and phase stability of the ISIS LLRF is 1º and 1% but higher stabilities would be welcome if possible.

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

2. Design description of the ISIS LLRF

The LLRF system basically consists of two parts. The first part is the hardware and FPGA program which regulates the I and Q components of the measured cavity voltage, hence the amplitude and phase of the cavity field and the second part performs cavity tuning to eliminate the reflected power. These two parts are explained in more details in the following sub-sections:

2.1. Amplitude and phase loops

A simplified design of the ISIS LLRF (amplitude and phase loops) is shown in Figure 1:

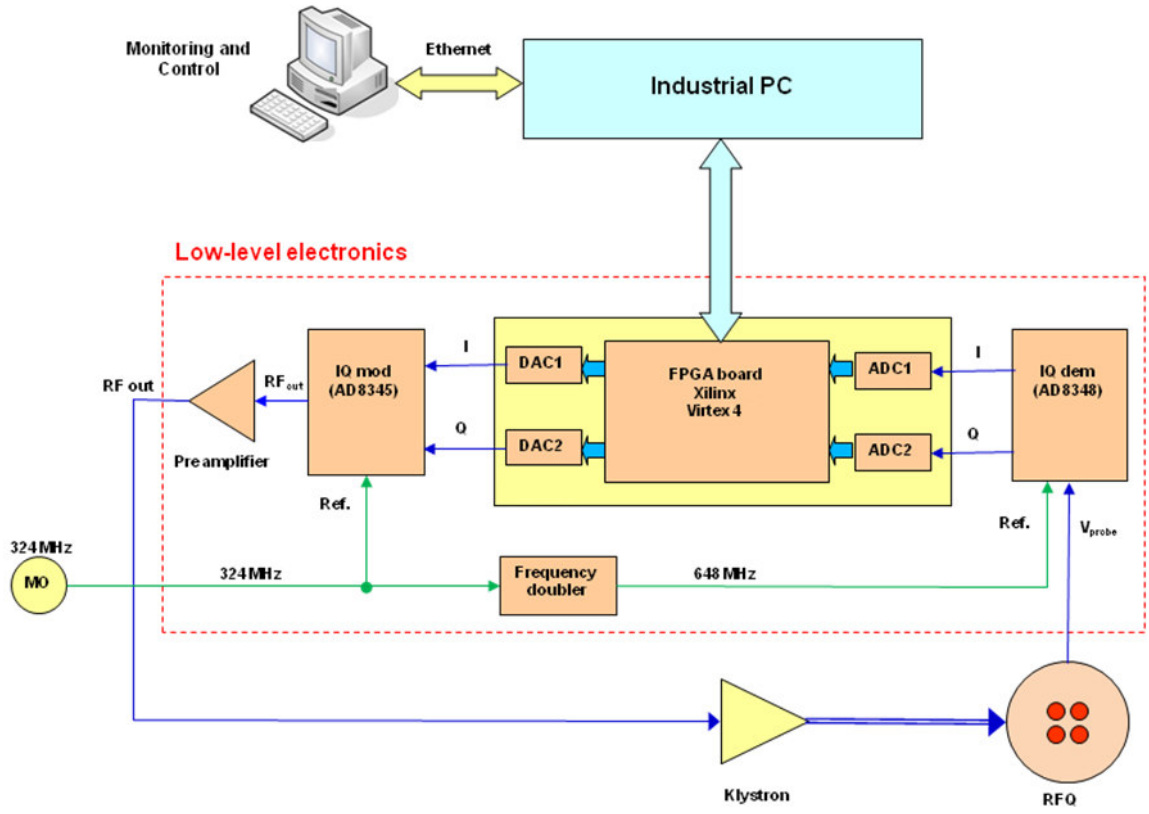

Figure 1: Simplified design of the ISIS LLRF (Amplitude and phase loops)

In this design an IQ demodulator board is used to convert the measured RFQ field into baseband I and Q (In-phase and Quadrature-phase). The resultant I and Q outputs are in the next stage converted from differential to single ended and conditioned so that they meet the input requirements of the two ADCs. The I and Q signals are then sampled and fed into the FPGA board for all kinds of processing which are needed to be done on the baseband signals, for example: rotation of the IQ vector (compensation of loop delays),

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

addition of fixed DC values to the I and Q inputs (compensation of DC offsets), PI regulation and addition of feed-forward signals (compensation of predictable errors, also for open-loop operation). At the output of the FPGA the I and Q signals are converted to analog by the two DACs and then are fed into the IQ modulator generating the drive for the RF amplifier. The LLRF system can be controlled and monitored by a local computer running under MATLAB-Simulink. This option, however, is mainly intended for test purposes. For the final installation, the LLRF system should be integrated into the global control system of the accelerator facility.

2.2. Tuning loop

Figure 2 shows the design of the tuning loop which is also based on IQ demodulation:

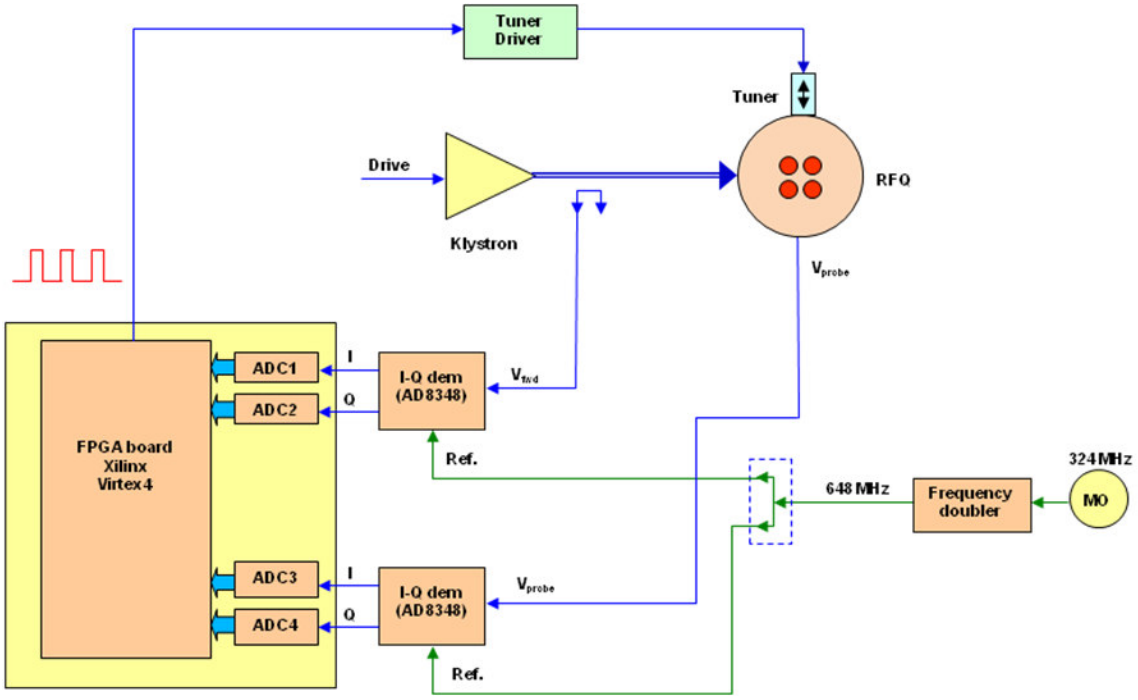

Figure 2: Simplified design of the ISIS LLRF (tuning loop)

In this design, the cavity probe voltage and the forward voltage are measured by two IQ demodulator boards and the resultant I and Q signals are sampled and fed into the digital board which is based on a FPGA (this board is physically the same as the one used for amplitude and phase regulation). The FPGA board calculates the phase difference between these two RF signals and depending on its value sends the required pulse and direction signals to the tuner driver to have the cavity matched to the klystron thus minimize the reflected power.

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

The tuning loop tends to keep the phase error (i.e. the phase difference between the forward and cavity voltages) between an upper and a lower threshold (typically a fraction of a degree). If, for example, the measured error exceeds the upper threshold, the loop moves the tuner in the right direction until the error becomes smaller than the upper threshold and then leaves the tuner at that position until the next time the phase error exceeds any of the thresholds.

In order to prevent the tuner from moving continuously in and out, tuning is done only during the RF pulse after the phase has settled. For that purpose the tuning loop is synchronized with the 50 Hz RF pulse rate through the FPGA program.

3. Implementation of the ISIS LLRF

3.1. LLRF rack

A 19-inch industrial rack (not part of the delivery) with a height of 2 m was used for the installation of the low-level electronics at the UPV-EHU RF lab. This rack accommodates the LLRF system comprising the Power Supply Unit, RF Distribution Unit, Analog Front-end Unit, Tuning Unit, FPGA Unit and the Tuner Driver Unit. The rest of the rack space was used for some other equipments (ex: RF generator, Laptop computer, oscilloscope, etc) during the system development and test. Figure 3 shows the LLRF rack with the units installed in it (the Tuner Driver Unit is not shown in the picture but it will be part of the delivery). Each of these units is explained in more details in the following subsections.

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

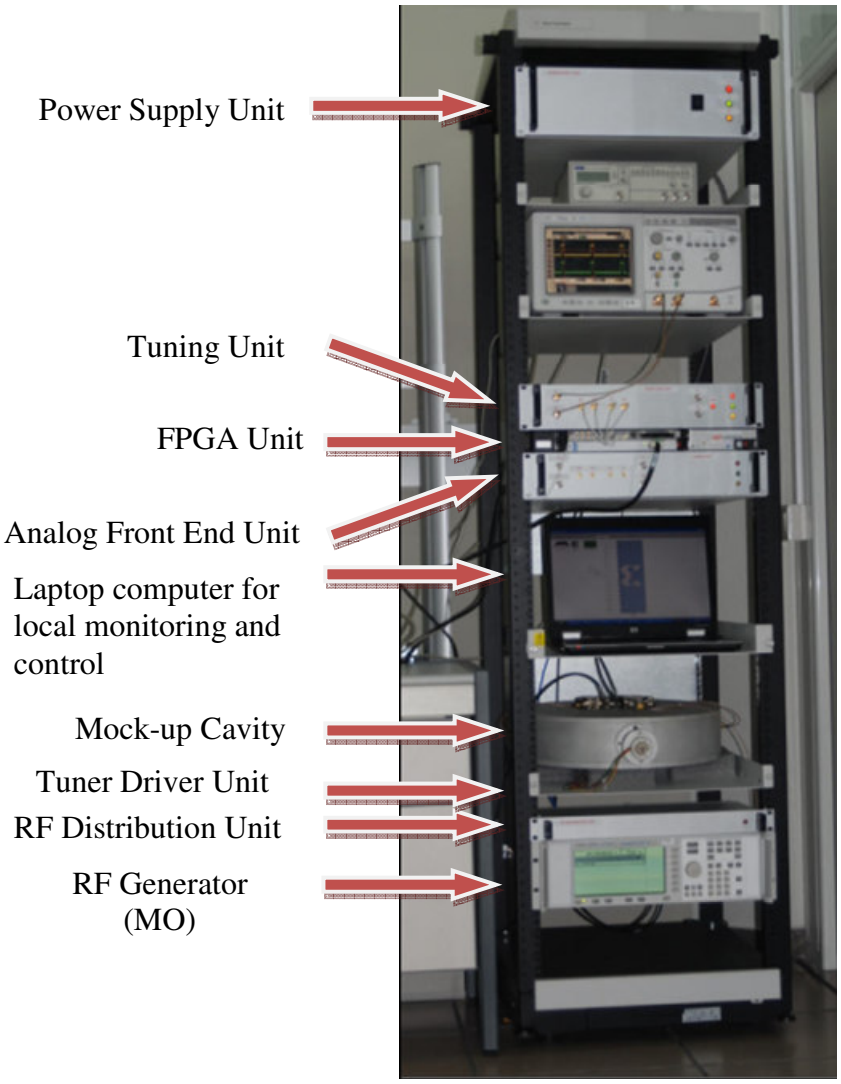

Figure 3: the LLRF rack

3.2. Power Supply Unit

The Power Supply Unit (3U) generates the required supply voltages (±5 V, +12 V) for all the other LLRF units. These are linear type power supplies from Power-one® with very low output noise and ripple.

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

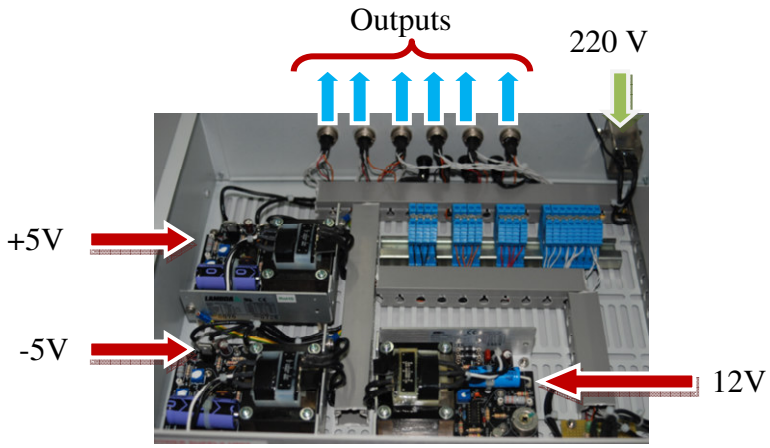

Figure 4: Picture of the interior of the Power Supply Unit

The LEDs on the front panel can be used to check the status of the output voltages with the following color code (The same color code has been used for the other units as well).

- Red LED: +5V
- Green LED: -5V
- Yellow LED: 12V

The output voltages are distributed to the other units via the six circular connectors mounted on the rear panel.

3.3. RF Distribution Unit

The output signal of the RF generator (Master Oscillator) is fed into the RF Distribution Unit with a typical level of 15dBm. After frequency doubling and filtering, this unit distributes both f_{RF} and $2*f_{RF}$ among the other units. While f_{RF} is needed for the IQ modulator, $2*f_{RF}$ is used for the IQ demodulators included in the regulation loops or for RF diagnostics. The components for the implementation of the RF Distribution Unit are mainly purchased from Mini-circuits.

The schematic of this unit is shown in Figure 11.

Figure 5 shows a picture of the interior of the RF Distribution Unit:

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

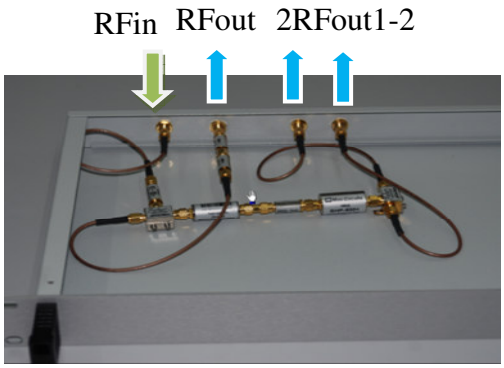

Figure 5: Picture of the RF Distribution Unit

The interconnection of the RF Distribution Unit with the other units is summarized in Table 2:

Table 2: RF Distribution Unit interconnections

Port	**Level (dBm)**	**In /Out**	**Function**
RFin	15	input	324 MHz ref. from the master oscillator (MO)
RFout	-2	output	324 MHz reference for the IQ modulator of the Analog Front-end Unit
2RFout-1	-7	output	648 MHz reference for the IQ demodulator of the Analog Front-end Unit
2RFout-2	-7	output	648 MHz reference for the two IQ demodulators of the Tuning Unit

The power budget of the RF Distribution Unit is summarized in Table 3:

Table 3: Power budget estimation of the RF Distribution Unit

Component	**Power Level**	**Units**	**Comment**
RFin (MO)	15	dBm	
Gain_VLF-320	-1	dB	
Gain_splitter1	-3	dB	
Gain_attenuator	-13	dB	
Gain_doubler	-13.5	dB	
Gain_VLF-630	-1.2	dB	
Gain_HPF	-0.4	dB	
Gain_splitter2	-3	dB	
RFout	-2	dBm	Abs. max (IQ mod) = 10 dBm

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

2RFout1	-7	dBm	Abs. max (IQ dem) = 10 dBm
2RFout2	-7	dBm	Abs. max (IQ dem) = 10 dBm

3.4. Analog Front-End Unit

This unit includes the electronics which are needed for RF-to-baseband and baseband-to-RF conversion. It's based on an AD8348 Evaluation Board from Analog Devices® for IQ demodulation, an in-house developed board based on AD8130 to convert the I/Q outputs of the demodulator from differential to single-ended, another in-house developed board based on AD8132 to convert from single-ended to differential and an AD8345 Evaluation Board for IQ modulation. The circuit diagrams of the converter boards are included in the schematics folder.

The complete schematic of the unit is shown in
Figure 12. The following figure shows the interior of the unit.

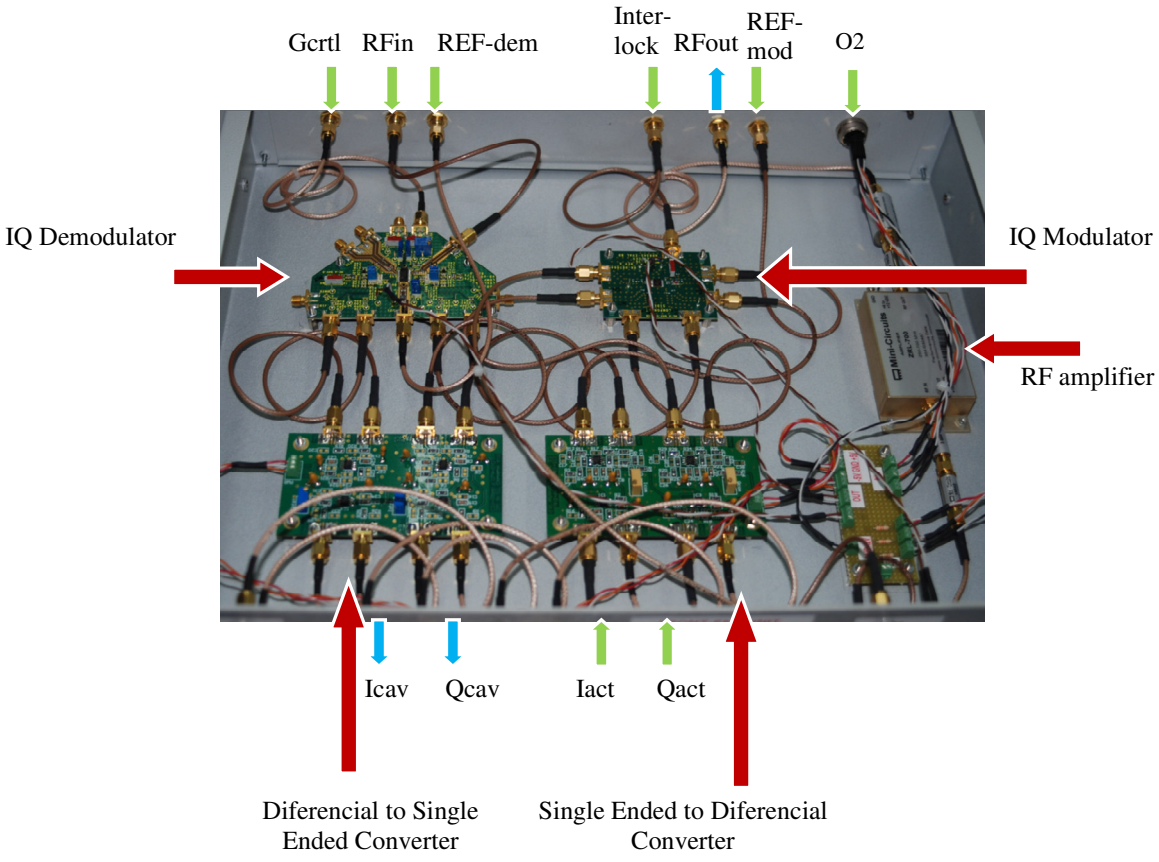

Figure 6: Picture of the Analog Front-End Unit

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

The input/output ports of the Analog Front-end Unit and their functions are explained in the next table.

Table 4: Input and output ports of the Analog Front–end Unit

Port	Level	Input/ Output	Function
O2	±5 V, 12V	input	Voltage supply
Gctrl	0.65 V (typ.)	input	Gain control of the probe signal demodulator
RFin	0 dBm (typ. max)	input	Cavity probe voltage
REF-dem	-10 dBm	input	RF reference signal for the demodulator. Frequency = $2*f_{MO}$
Interlock	0V / 5 V	input	IQ mod. disable (in case of fault)
RFout		output	drive signal for the RF amplifier supplying the cavity
REF-mod	-10 dBm	input	RF reference signal for the modulator. Frequency = f_{MO}
Icav	±1.4 V (max)	output	I component of the cavity probe signal (SMA)
Qcav	±1.4 V (max)	output	Q component of the cavity probe signal (SMA)
Icav	±1.4 V (max)	output	I component of the cavity probe signal (BNC)
Qcav	±1.4 V (max)	output	Q component of the cavity probe signal (BNC)
Iact	±0.6 V (max)	input	I component of the cavity drive signal (SMA)
Qact	±0.6 V (max)	input	Q component of the cavity drive signal (SMA)
Iact	±0.6 V (max)	output	I component of the cavity drive signal (BNC)
Qact	±0.6 V (max)	output	Q component of the cavity drive signal (BNC)

The cavity probe signal (to be regulated by the LLRF) is fed into the Analog Front-end Unit via the RFin port. A gain control input voltage should be provided for the IQ demodulator through the Gctrl port. As an alternative, the demodulator gain can be adjusted by the corresponding potentiometer mounted on the demodulator board and setting the SW13 switch to the POT position. This, however, is not the preferred manner of gain adjustment as potentiometers are in general subjected to drifts. In the current configuration, one of the DAC channels is dedicated to gain adjustment of the IQ demodulator. The level of the gain control voltage can be between 0.2 V and 1.2 V but it's normally fixed at 0.65V app. via the control computer as that gives best results in terms of output linearity versus gain control voltage. The reference RF ($f=2*f_{RF}$) for the IQ demodulator is provided by the RF Distribution Unit with a typical level of -10 dBm. The I and Q outputs of the IQ demodulator, after being converted from differential to single-ended by the corresponding converter board are sent to the SMA connectors on the

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

front panel to be sampled and fed into the FPGA for the subsequent signal processing. Additionally, a buffered copy of the I and Q signals is available on the BNC connectors on the front panel so that they could be measured by an oscilloscope without disturbing the feedback loops. The maximum swing of the I and Q signals is ±1.4 V but care should be taken in order not to saturate the ADC inputs when these signals are fed into the FPGA unit (the maximum input voltage of the ADCs is only ±1.25V). This can be done by inserting appropriate attenuators on the RFin of the IQ demodulator and/or fine adjustment of the demodulator gain about 0.65V.

The I and Q inputs of the IQ modulator (the I/Q drive signals) are fed into the corresponding SMA connectors on the front panel with a buffered copy available on the BNC connectors for local monitoring. The maximum voltage of the I/Q inputs is ±0.6 V but that whole voltage range might not be used as the DACs have a more limited voltage range (it's only ±0.32 V). The I and Q signals are in the next stage converted from single-ended to differential and fed into the IQ modulator with the RF reference being provided by the RF Distribution Unit. Similar to the IQ demodulator, appropriate attenuators should be placed at the RF output of the LLRF so that no LLRF signal is saturated within the whole power range of the RF amplifier supplying the cavity. The RF output of the IQ modulator is amplified by a ZLR-700 RF amplifier from Mini-circuits® with maximum output power of 25dBm. However, in order to make sure that the amplifier will always work in its linear region, some attenuators have been placed at the amplifier input; therefore the actual output power of the Analog Front-end Unit will be less than the rated power of the amplifier.

An additional port has been provided on the rear panel for enabling/disabling the IQ modulator for interlock purposes with a typical turn-off time of 1.5 µs. The IQ modulator can also be manually enabled and disabled using the corresponding switch mounted on it. An LED on the front panel indicates whether the modulator is enabled or disabled.

3.5. Tuning Unit

Figure 7 shows a picture of the Tuning Unit.

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

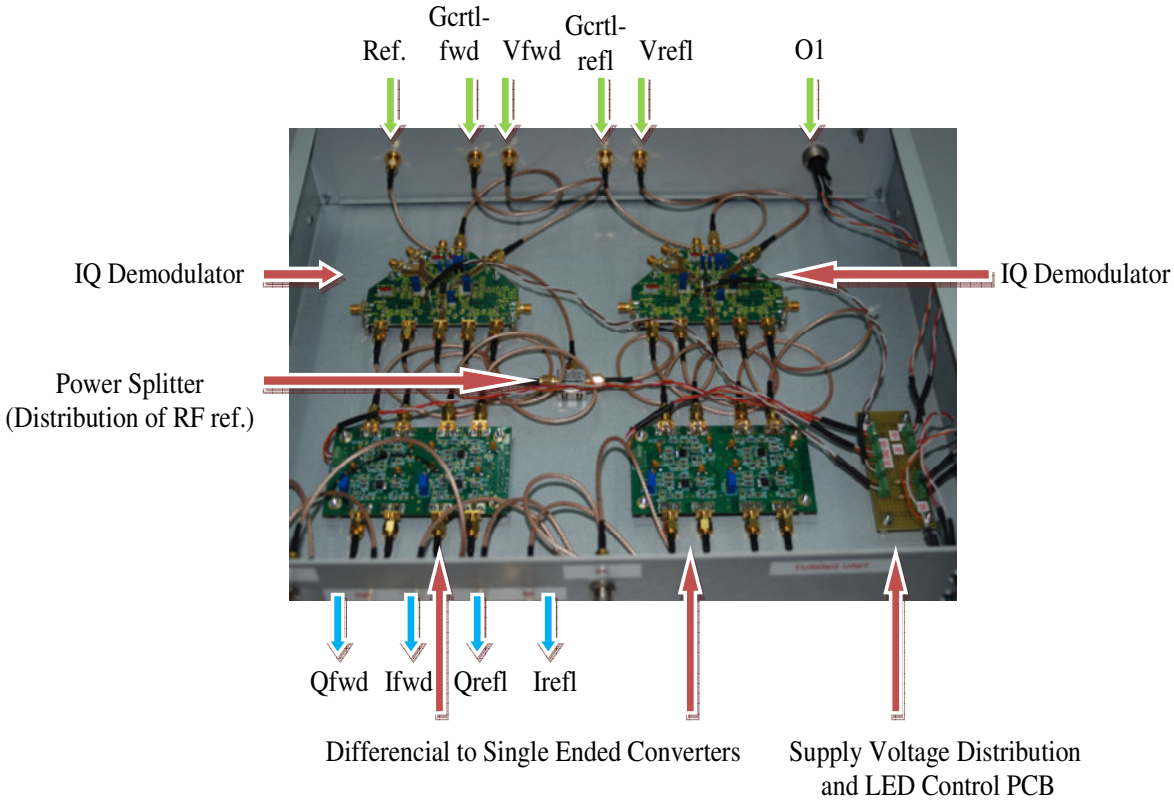

Figure 7: Picture of the Tuning Unit

This unit basically consists of two AD8348 IQ demodulator boards plus two in-house-developed boards, similar to the one used in the Front End Unit, to convert the differential outputs of the IQ demodulators to single ended. The forward voltage and the cavity reflected voltage are fed into the unit via the corresponding connectors on the rear panel to be converted to baseband. The reference RF ($2*f_{RF}$) which is provided by the RF Distribution Unit is split into two by a power splitter in the Tuning Unit and fed into the two boards. The I/Q outputs of the two demodulators are then converted to single-ended by the two converter boards and made available on the corresponding SMA connectors on the front panel to be sampled by the ADCs. The voltage range of these signals is similar to the one of the Analog Front-end Unit. Likewise, a buffered copy of the I/Q signals is additionally sent to the BNC connectors on the front panel for local monitoring. The four baseband outputs of the Tuning Unit are then sampled and fed into the FPGA running the tuning program.

Two SMA connectors are mounted on the rear panel so that the user can adjust the gain of the IQ demodulators (the preferred voltage though is 0.65V). Alternatively, the gains can be adjusted using the corresponding potentiometers and switches mounted on the demodulator boards. It should be noted that while the Vfwd demodulator is needed for the tuning loop, the Vrefl one is only used for monitoring the value of the reflected voltage in the control computer. For that reason, in the current version of the FPGA

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

program, a DAC channel is dedicated to the gain adjustments of the Vfwd demodulator but the gain of the Vrefl demodulator is manually fixed at 0.65V app. by the gain adjustment potentiometer. The schematic of the Tuning Unit is shown in fig. 13.

The next table summarizes the input and output ports of the Tuning Unit and their functions:

Table 5: Tuning Unit ports and their functions

Port	Level	In/ Out	Function
O1	±5 V, 12V	input	Voltage supply
Ref	-7 dBm	input	RF reference for both demodulators; frequency = $2*f_{MO}$
Vfwd	0 dBm (typ max)	input	Cavity forward voltage
Vrefl	0 dBm (typ max)	input	Cavity reflected voltage
Gctrl_fwd	0.65 V (typ.)	input	Gain control voltage of the Vfwd demodulator
Gctrl_refl	0.65 V (typ.)	input	Gain control voltage of the Vrefl demodulator
Ifwd	±1.4 V (max)	output	I component of the cavity forward voltage (SMA)
Qfwd	±1.4 V (max)	output	Q component of the cavity forward voltage (SMA)
Ifwd	±1.4 V (max)	output	I component of the cavity forward voltage (BNC)
Qfwd	±1.4 V (max)	output	Q component of the cavity forward voltage (BNC)
Irefl	±1.4 V (max)	output	I component of the cavity reflected voltage (SMA)
Qrefl	±1.4 V (max)	output	Q component of the cavity reflected voltage (SMA)
Irefl	±1.4 V (max)	output	I component of the cavity reflected voltage (BNC)
Qrefl	±1.4 V (max)	output	Q component of the cavity reflected voltage (BNC)

3.6. Tuner Driver Unit

This unit consists of the stepper motor driver for cavity tuning and an in-house-developed board which conditions the DAC output signals (Pulse and Direction) to have TTL level as this is required for the current driver.

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

The stepper motor driver is an Easy Step ™ 3000 Industrial Interface from Active Robots® (please refer to its corresponding datasheet in the datasheets folder for more information). The stepper motor of the mockup cavity used during the LLRF tests was Mclennan 26DBM series DLA´s.

Figure 8 shows the Tuner Driver Unit with the driver and conditioning boards mounted in it:

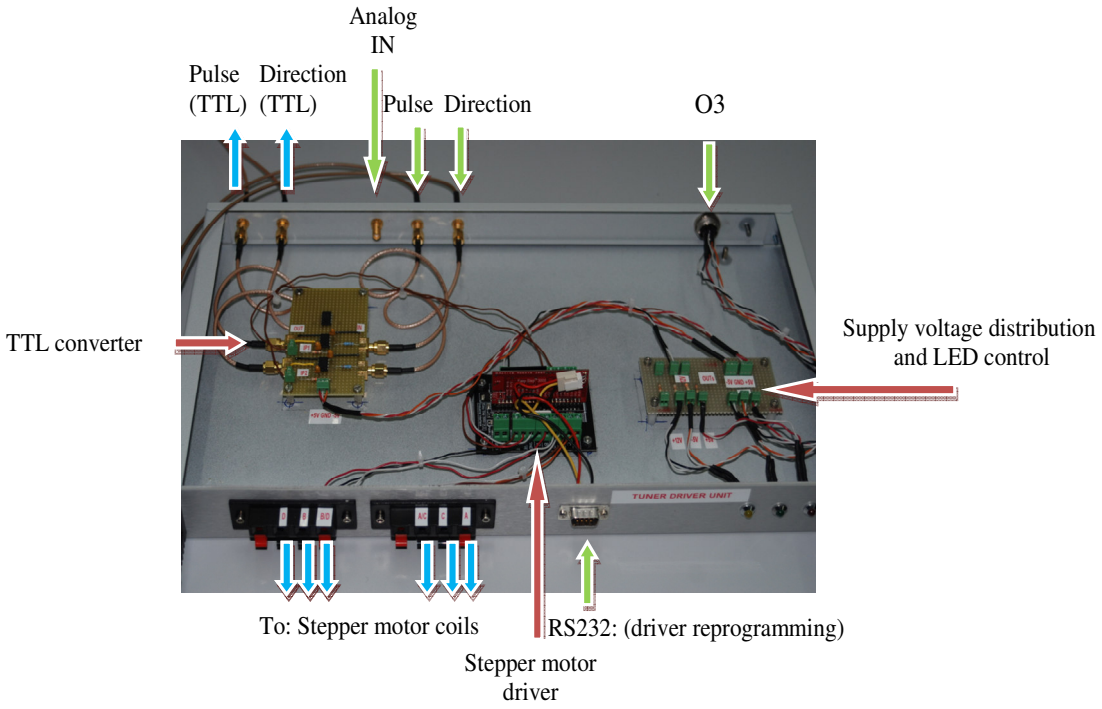

Figure 8: Picture of the Tuner Driver Unit

The pulse and direction inputs are fed into the Tuner Driver Unit via the corresponding connectors on the rear panel. The level of these signals is fixed in the FPGA program so that the conditioning board generates the required signal levels for the output ports and the driver board. An analog input port is additionally foreseen so that the driver can also work in analog input mode if needed (with the current design, though, this port is not used). The pulse and direction signals with TTL levels are available on the corresponding connectors on the rear panel. These signals can therefore be used if the Tuner Driver Unit should control a stepper motor not compatible with the current driver board (in that case only the TTL converter board of the Tuner Driver Unit will be used as the stepper motor should be driven by another stepper motor driver). The required pulses for the stepper motor coils are accessible via the two quick connect terminals on the front panel. Also, an RS 232 port has been foreseen on the front panel to reprogram the driver board if needed (please refer to the Easy Step ™ 3000 Industrial Interface datasheet for details)

Table 6 shows the input and output ports of the Tuner Driver Unit and their functions:

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

Table 6: Input/output ports of the Tuner Driver Unit and their functions

Port	Level	In/ Out	Function
O3	±5 V, 12V	input	Voltage supply
Analog IN	-	input	Not used in the current design
Pulse	0V / 0.2V	input	Pulse input for the driver board (from the Digital Unit)
Direction	0V / 0.2V	input	Direction input for the driver board (from the Digital Unit)
Pulse (TTL)	TTL	output	Pulse output with TTL level
Direction (TTL)	TTL	output	Direction output with TTL level
A,C,AC, B,D,BD		output	Current pulses for the stepper motor coils
RS-232		input	Programming the driver board

3.7. Digital Unit

The Digital Unit includes the FPGA board and the ADC and DAC cards. This is a high performance VHS-ADC board from Lyrtech® (with cPCI to 4x PCIe interface for PC connection) based on a Virtex4 FPGA from Xilinx and eight AD6645 ADCs with a resolution of 14 bits and maximum sampling rate of 105 MSPS (please refer to the corresponding datasheets about these boards for more information). The board also incorporates 128 MB of SDRAM for data buffering on the FPGA. In order to provide the FPGA board with 8 additional DAC inputs, an add-on DAC module from Lyrtech was also purchased and mounted on the FPGA board. These are DAC5687 from Texas Instruments with a resolution of 14 bits and maximum sampling rate of 480 MSPS (interpolating).

The FPGA was programmed using the Xilinx® System Generator and the Lyrtech® Model Based Design Kit in the MATLAB-Simulink environment. The advantage of this method was that the FPGA programming was eased and the time needed to develop the FPGA code was significantly reduced compared to VHDL.

In order to further reduce the time and effort needed for the establishment of the LLRF test bench, it was decided to co-simulate the FPGA hardware in the MATLAB-Simulink environment. With this method, known as HIL (Hardware-In-the-Loop) co-simulation, the FPGA communicates directly with Simulink blocks; therefore the FPGA hardware can be tested without having to connect it to real-world signals. Then, Simulink source blocks were used to set the LLRF input parameters (such as the PI gains and the offset values) eliminating the need for an additional GUI (Graphical User Interface) and a communication protocol, while the I/Q input and output signals were still the real ones

coming from the Analog Front-end Unit connected to the mockup cavity. It should be noted, however, that although the HIL co-simulation of the loop proved to be very useful during the system development, test and trouble-shooting, it's not planned to use it for the final deployment as the LLRF system should be integrated into the global control system of the accelerator facility.

Figure 9 shows the FPGA card cage with the cPCI port, the VHS-ADC board and the DAC module mounted in it. For information on these items, please refer to the corresponding datasheets in the Datasheets folder.

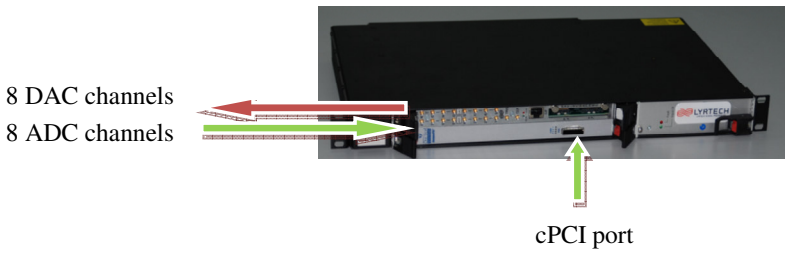

Figure 9: Picture of the Digital Unit

3.8. PC Unit

A laptop computer (HP EliteBook 8530p, 2.4 GHz with 2 GB of RAM and 250 GB of HD) was used for the development of the FPGA program and later for the LLRF tests with the mockup cavity. As this laptop is not planned to be part of the delivery, it's important to consider the following requirement from Lyrtech® when a new laptop for the LLRF system is being purchased.

The FPGA interface with the laptop might malfunction if any of these requirements is not met:

- Manufacturer: Dell or Hewlett Packard (HP)
- Operating System: Microsoft Windows XP Professional
- Service Pack 2
- Express Card for compact PCI connection

Installation of the following software packages on the laptop is required before the FPGA can be programmed or the HIL co-simulalation can run:

- Software for BSDK: ISE Foundation 9.2
- Software for MBDK: Matlab R2007a and System Generator for DST 9.2

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

For detailed information on the required software packages and the installation procedure, please refer to the corresponding User's Manual from Lyrtech®.

3.9. Mock-up Cavity

A mock-up cavity was designed and manufactured in order to perform the LLRF tests at low power and validate the performance of the regulation loops. The cavity is of re-entrant type to reduce the size, and is made from Aluminum except the plunger which is made of steel. Its resonance frequency is compliant with the ISIS requirements (i.e. 324 MHz) and its quality factor was measured at 1450 approximately. The cavity was designed by Ellyt® and manufactured by Tekniker®. Table 7 and Figure 10 summarize the results of the cavity measurements with a Network Analyzer with the tuner moved to both ends:

Table 7: Resonant cavity measurements

f_{cavity} (MHz)	Plunger position	IL(dB)	Q_L
325.7750	Completely out	-15.12	1448
326.9000	Completely in	-16.15	1454

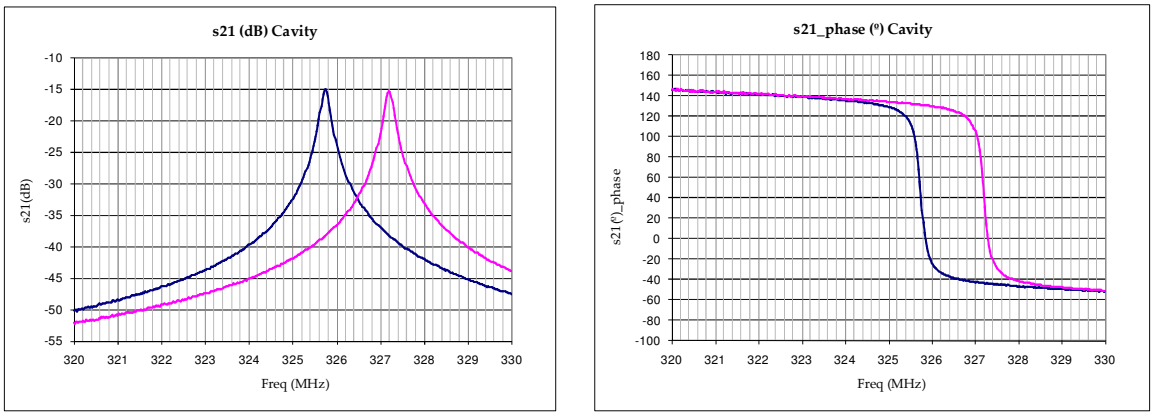

Figure 10: Resonant cavity S21 measurements

4. LLRF unit schematics

Schematics of the LLRF units explained earlier are shown in the next figures:

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

4.1. RF Distribution Unit

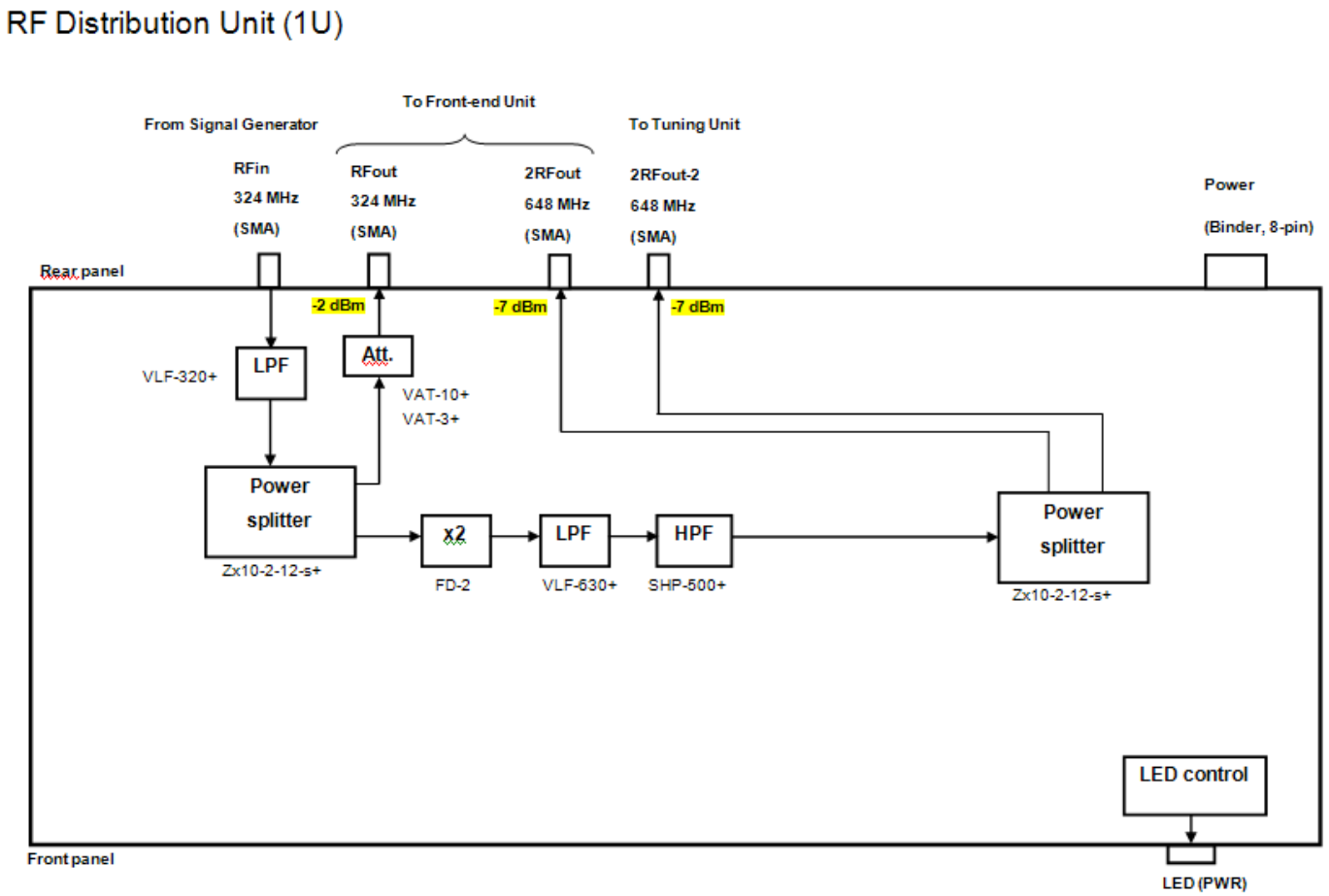

Figure 11: Schematic of the RF Distribution Unit

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

4.2. Analog Front-End Unit

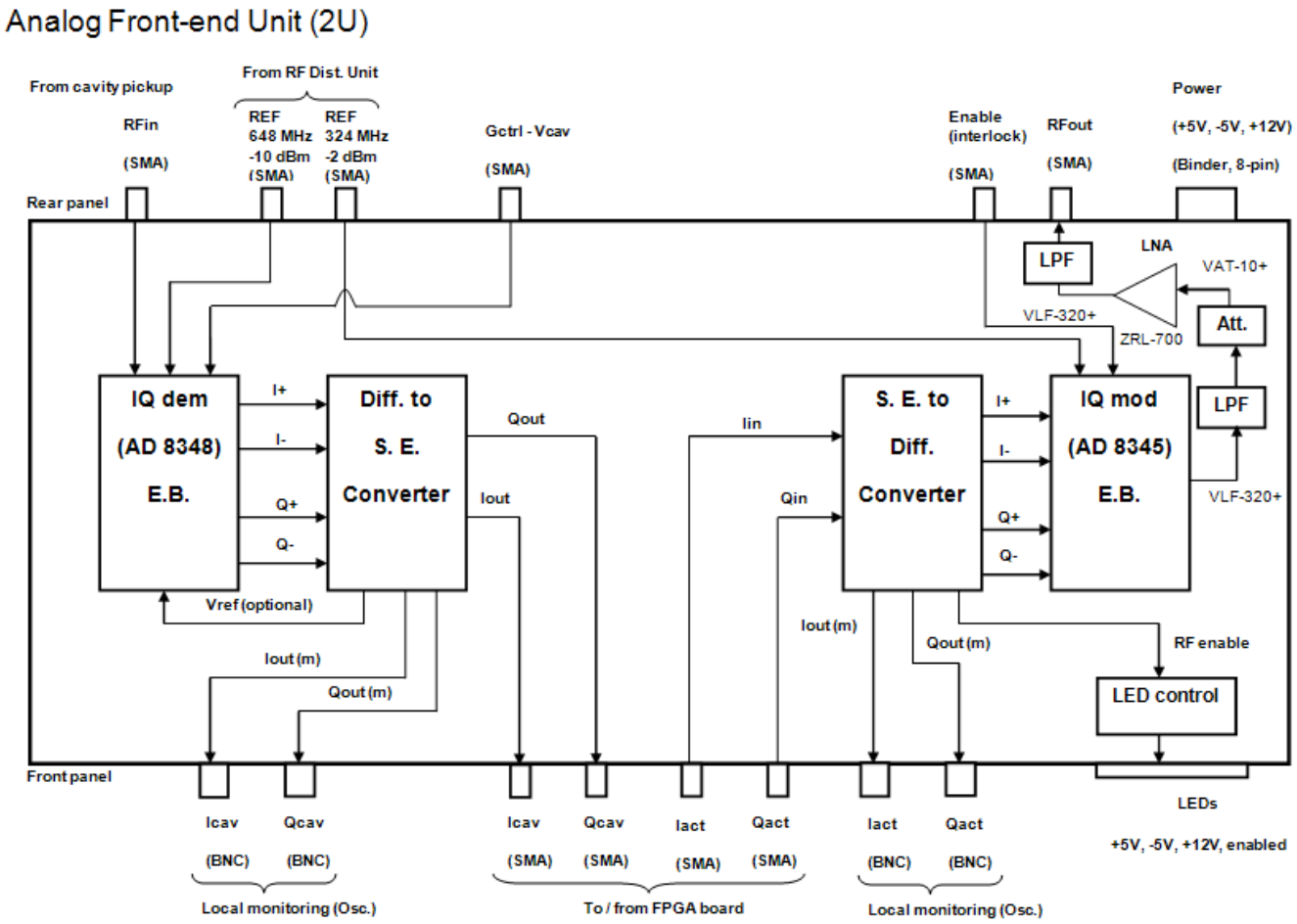

Figure 12: Schematic of the Analog Front-End Unit

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

4.3. Tuning Unit

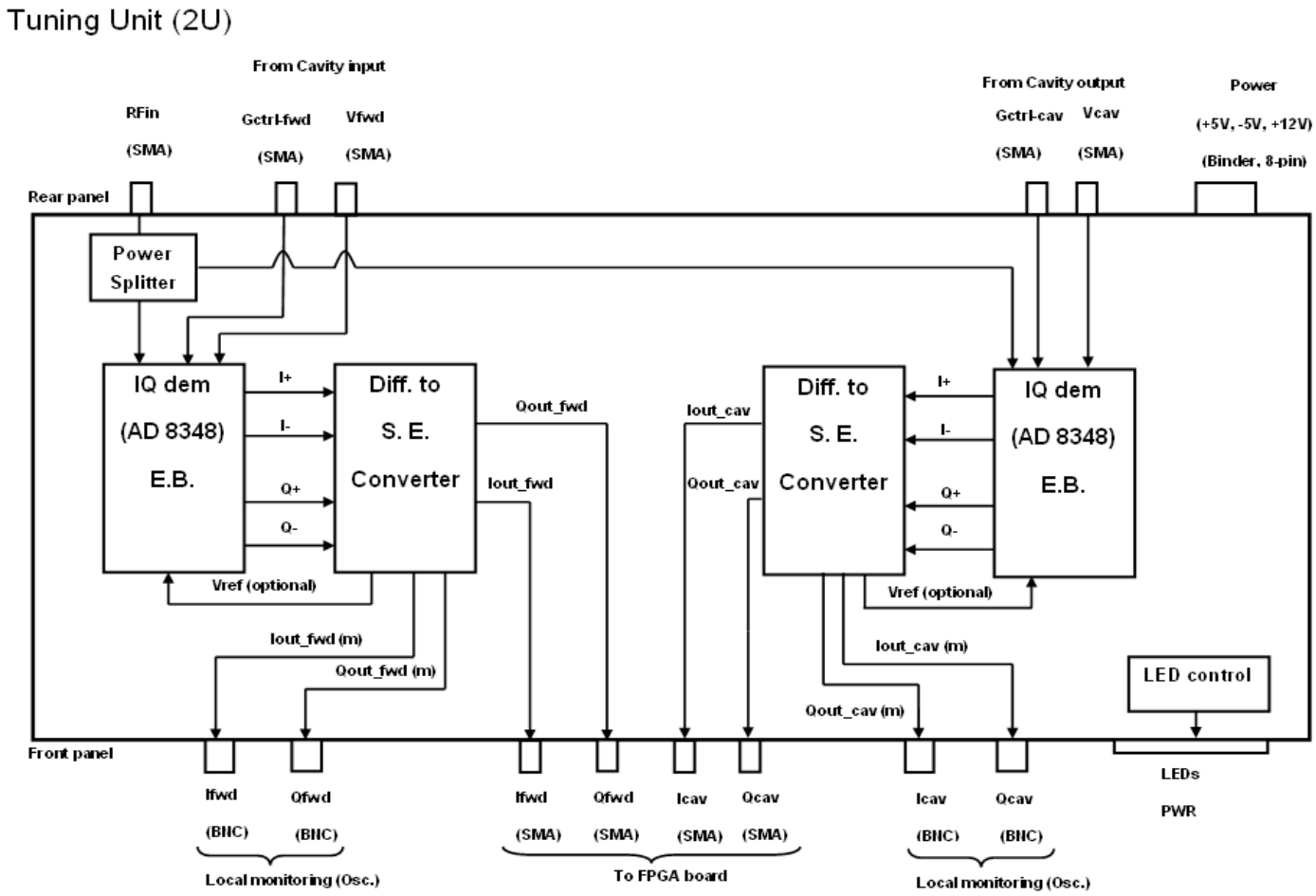

Figure 13: Schematic of the Tuning Unit

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

4.4. Tuner Driver Unit

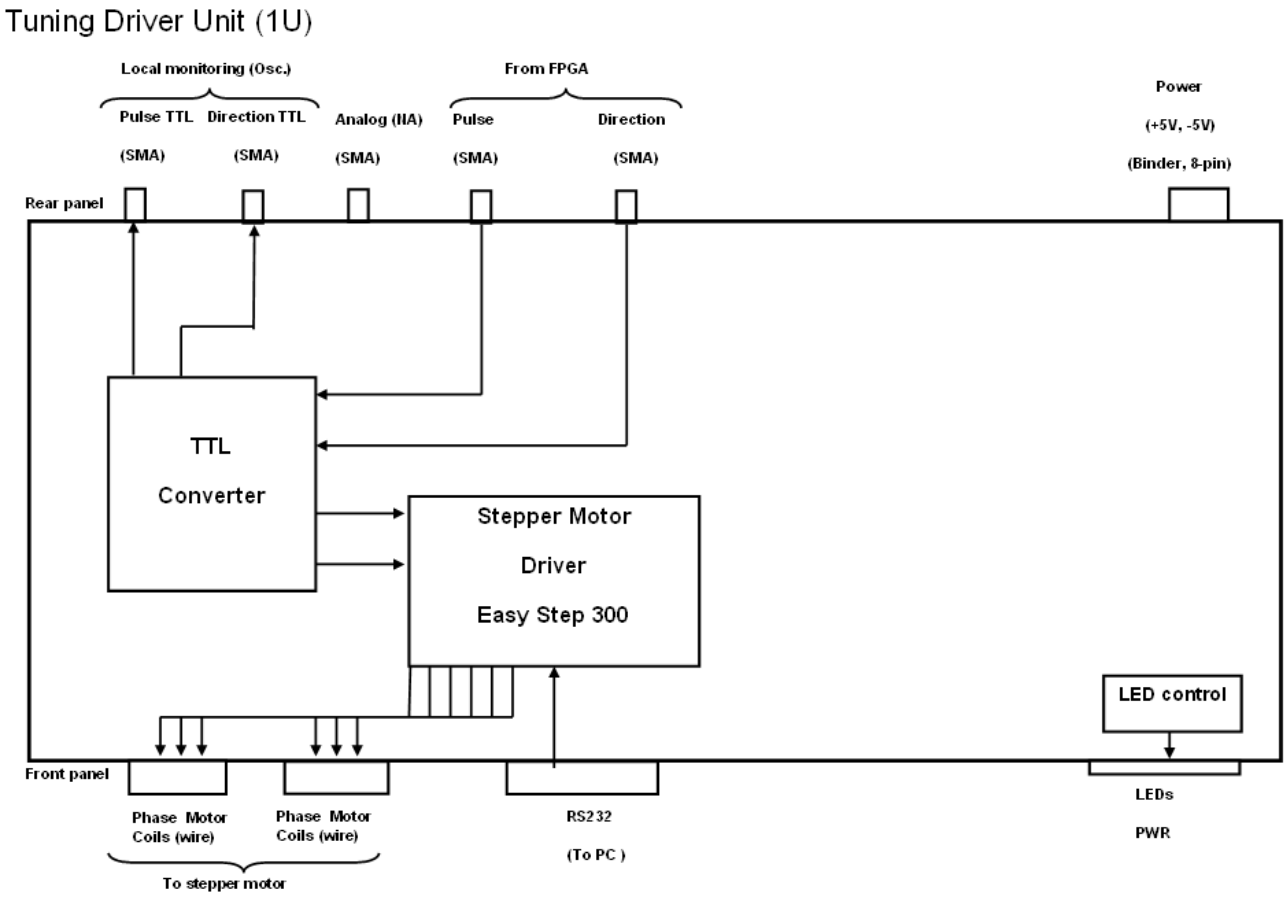

Figure 14: Schematic of the Tuner Driver Unit

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

5. Unit interconnections

The interconnections among the different LLRF units are summarized in the following tables:

Table 8: Power Supply Unit Interconnections

Name	Input/ output	Panel Position	Connection Type	From/To Unit	Operating Level	Abs_Max. Rating*
Power inlet	input	rear	220 V inlet	From power line	220 V	
O1	output	rear	Binder 8pin	To Tuning Unit (*connection with cable O1!*)	±5 V, 12 V	±5.5 V
O2	output	rear	Binder 8pin	To Analog Front-end Unit (*connection with cable 02!*)	±5 V, 12 V	±5.5 V
O3	output	rear	Binder 8pin	To Tuner Driver Unit (*connection with cable O3!*)	±5 V, 12 V	
O4	output	rear	Binder 8pin	Spare	±5 V, 12 V	
O5	output	rear	Binder 8pin	Spare	±5 V, 12 V	
O6	output	rear	Binder 8pin	Spare	±5 V, 12 V	
+5 V	NA	front	LED red	Not Applicable (NA)	On	
-5 V	NA	front	LED green	NA	On	
+ 12 V	NA	front	LED yellow	NA	On	

Note:

(1) Three outputs of the Power Supply Unit are foreseen as spares in case they will be needed in the future for other units.
(2) The abs_max ratings refer to the IQ mod. and dem. boards.

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

Table 9: RF Distribution Unit Interconnections

Name	Input/ output	Panel Position	Connection Type	From/To Unit	Operating Level	Abs_Max. Rating*
RFin	input	rear	SMA	From RF Signal Generator	15 dBm	
RFout	output	rear	SMA	To Ref-mod of the Analog Front-end Unit	-2 dBm	10 dBm
2RFout-1	output	rear	SMA	To Ref-dem of the Analog Front End Unit	-7 dBm	10 dBm
2RFout-2	output	rear	SMA	To Ref of the Tuning Unit	-7 dBm	10 dBm
	NA	front	LED red	NA	Off	

** Note: These abs_max ratings refer to the IQ mod/dem boards of the Analog Front-end Unit and Tuning Unit

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

Table 10: Analog Front End Unit Interconnections

Name	Input/ output	Panel Position	Connection Type	From/To Signal Unit	Operating Level	Abs_Max. Rating
O2	input	rear	Binder (8 pin)	From Power Supply Unit	±5 V, 12 V	
RFin	input	rear	SMA	From cavity pickup loop	0 dBm (max)	10 dBm
Ref-dem	input	rear	SMA	From 2RFout-1 of RF Distribution Unit	-7 dBm	10 dBm
Gcrtl	input	rear	SMA	From channel 16 of the Digital Unit	0.2 V–1.2 V	5.5 V
Ref-mod	input	rear	SMA	From RFout of the RF Distribution Unit	-2 dBm	10 dBm
Interlock	input	rear	SMA	From interlock system (optional)	TTL	
RFout	output	rear	SMA	To RF amplifier	To be set	
Icav	output	front	SMA	To channel 3 of the Digital Unit	±1.4 V (max)	
Icav	output	front	BNC	To local oscilloscope	±1.4 V (max)	
Qcav	output	front	SMA	To channel 4 of the Digital Unit	±1.4 V (max)	
Qcav	output	front	BNC	To local oscilloscope	±1.4 V (max)	
Iact	input	front	SMA	From channel 12 of the Digital Unit	±0.6 V (max)	
Iact	output	front	BNC	To local oscilloscope	±0.6 V (max)	
Qact	input	front	SMA	From channel 13 of the Digital Unit	±0.6 V (max)	
Qact	output	front	BNC	To local oscilloscope	±0.6 V (max)	
+5 V	NA	front	LED		ON	
-5 V	NA	front	LED		ON	
+12 V	NA	front	LED		ON	

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

Table 11: Tuning Unit Interconnections

Name	Input/ output	Panel Position	Connection Type	From/To Signal Unit	Operating Level	Abs_Max. Rating
O1	input	rear	Binder (8 pin)	From the Power Supply Unit	±5 V, 12 V	
Ref	input	rear	SMA	From 2RFout-2 of the RF Distribution Unit	-7 dBm	10 dBm
Vfwd	input	rear	SMA	From directional coupler measuring the cavity forward voltage	0 dBm (max)	10 dBm
Gctrl-fwd	input	rear	SMA	From channel 15 of the Digital Unit	0.2 V – 1.2 V	5.5 V
Vrefl	input	rear	SMA	From directional coupler measuring the cavity reflected voltage	0 dBm (max)	10 dBm
Gctrl-refl	input	rear	SMA	NA (the gain adjustment is manual with the current configuration)	0.2 V – 1.2 V	5.5 V
Ifwd	output	rear	SMA	To channel 2 of the Digital Unit	±1.4 V (max)	
Ifwd	output	front	BNC	To local oscilloscope	±1.4 V (max)	
Qfwd	output	front	SMA	To channel 1 of the Digital Unit	±1.4 V (max)	
Qfwd	output	front	BNC	To local oscilloscope	±1.4 V (max)	
Irefl	output	front	SMA	To channel 5 of the Digital Unit	±1.4 V (max)	
Irefl	output	front	BNC	To local oscilloscope	±1.4 V (max)	
Qrefl	output	front	SMA	To channel 6 of the Digital Unit	±1.4 V (max)	
Qrefl	output	front	BNC	To local oscilloscope	±1.4 V (max)	
+5 V		front	LED		ON	
-5 V		front	LED		ON	
+12 V		front	LED	NA	OFF	

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

Table 12: Tuner Driver Unit Interconnections

Name	Input/ output	Panel Position	Connection Type	From/To Signal Unit	Operating Level	Abs_Max. Rating
O3	input	rear	Binder (8 pin)	From the Power Supply Unit	±5 V, 12 V	
Pulse TTL	output	rear	SMA	To final stepper motor driver (if needed)	TTL	
Direction TTL	output	rear	SMA	To final stepper motor driver (if needed)	TTL	
Analog IN	input	rear	SMA	NA		
Pulse	input	rear	SMA	From channel 12 of the Digital Unit	Defined in the FPGA	
Direction	input	rear	SMA	From channel 11 of the Digital Unit	Defined in the FPGA	
A	output	front	wire	To stepper motor		
C	output	front	wire	To stepper motor		
AC	output	front	wire	To stepper motor		
B	output	front	wire	To stepper motor		
D	output	front	wire	To stepper motor		
BD	output	front	wire	To stepper motor		
RS232	input	front	D type (9 pin)	To PC (only to reprogram the driver if needed)	RS 232	
+5 V		front	LED		ON	
-5 V		front	LED		ON	
+12 V		front	LED	NA	OFF	

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

6. Implementation of the amplitude and phase regulation loops

The following figure shows the schematics of the FPGA program for amplitude and phase regulation:

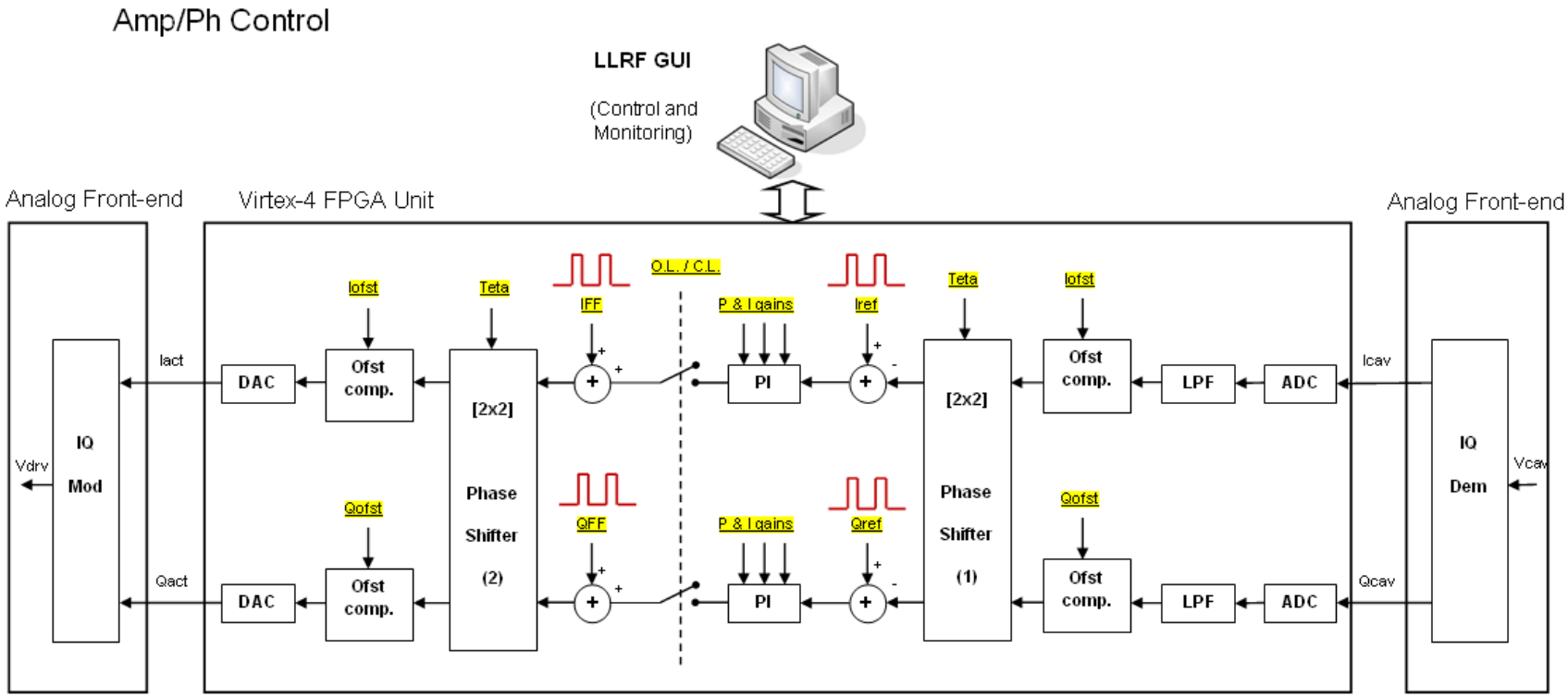

Figure 15: Simplified schematic of the FPGA program for amplitude and phase regulation; the parameters marked as underscore yellow are set from the control computer

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

As the figure shows, the measured cavity voltage (from the cavity pickup loop) is converted to baseband by an analog IQ demodulator board which is incorporated in the Analog Front-end Unit. The I and Q signals, after being converted from differential to single-ended, are sampled by 14 bit ADCs running at 104 MSPS and fed into a Virtex-4 FPGA for the required signal manipulation. That includes low-pass filtering for noise reduction, compensating the DC offsets of the IQ demodulator, baseband phase shifting, feed-forward control and PI regulation (please refer to the Parameter Setting section for information on how to adjust these parameters). The I and Q outputs of the FPGA are then converted to analog by 14 bit 480 MSPS (4x interpolation) DACs and fed into the front-end unit where they are used as the baseband inputs of the IQ modulator generating the drive for the RF amplifier.

The magnitude and phase of the RFQ field during the pulse is controlled by setting proper values for the *Iref* and *Qref* inputs from the control computer. The use of the feed-forward signals (*IFF* and *QFF*) is two-folded. First, they can be used to operate the cavity in open-loop mode (In this case, the *OL/CL* switch is opened; therefore the IQ modulator is only driven by the *IFF* and *QFF* inputs). This mode can be particularly useful for test purposes or for making sure that the regulation loops will be stable before they are actually closed. Second, it can be used to compensate for repetitive or predictable errors (such as the beam) before these errors are sensed and corrected by the I/Q regulation loops.

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

7. Implementation of the tuning loop

The following figure shows the schematics of the FPGA program for cavity tuning:

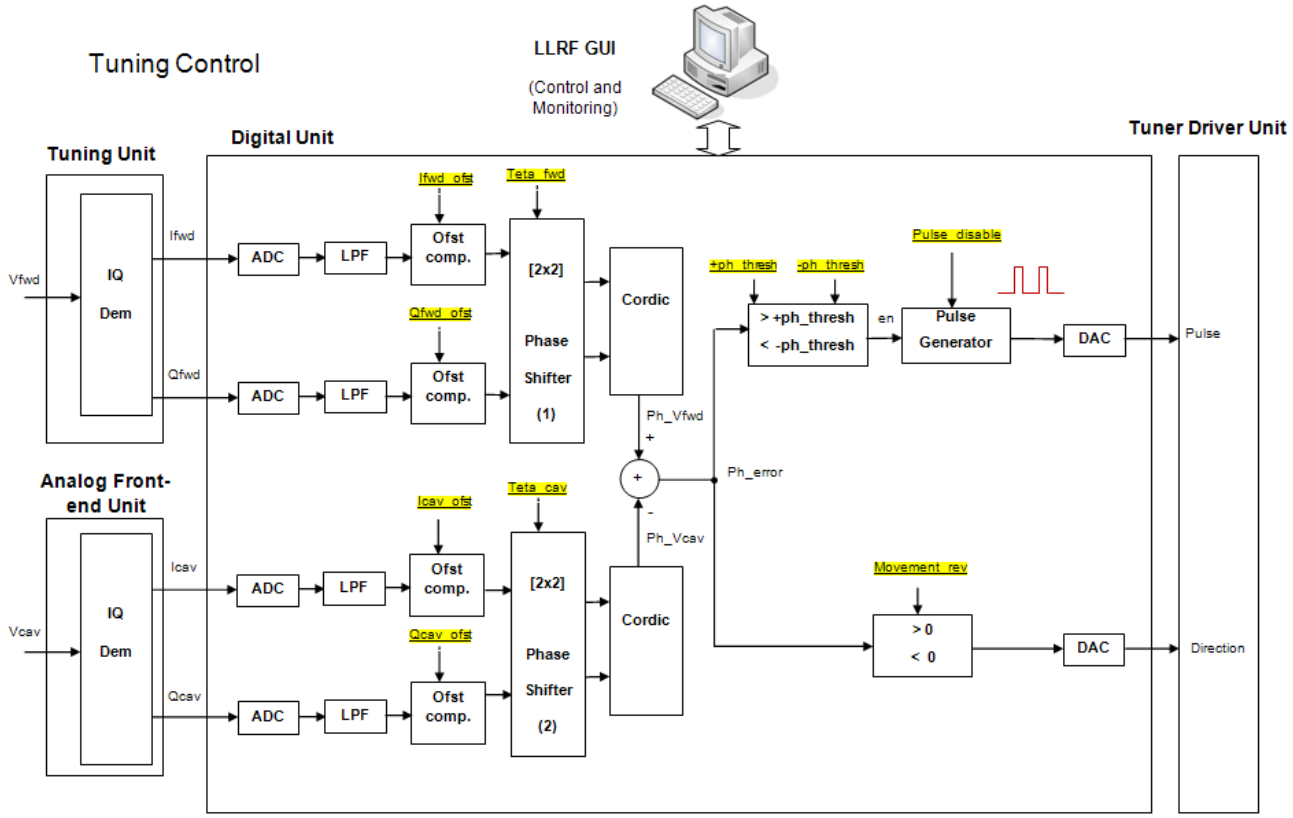

Figure 16: Simplified schematic of the FPGA program for the tuning loop

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

As the schematic show, the design of the tuning loop is also based on IQ demodulation. In this case, two IQ demodulators are used to convert the cavity forward and probe voltages to baseband (the IQ demodulator for the probe voltage is physically the same as the one used for amplitude and phase regulation) and the resultant I/Q signals are sampled and fed into the FPGA. Similar to the amplitude and phase loops, offset compensation blocks and low-pass filters have been used in the FPGA for offset compensation and noise reduction respectively (As the tuning loop is much slower than the amplitude and phase loops, the programmed low-pass filters for the tuning loop have a smaller bandwidth so that the noise on the base-band signals can be further reduced hence tuning can be done with more precision). In the next stage, the phase difference between the forward and cavity voltages is calculated by CORDIC blocks programmed in the FPGA. Then, the tuning loop tends to keep this phase difference as close as possible to its desired value where the desired phase is the one giving zero reflected power from the cavity with the presence of the beam. This is done by defining two phase thresholds (typically a fraction of a degree) above and below zero and keeping the actual phase error always between these thresholds by moving the tuner inwards/outwards if the error exceeds either of the thresholds. For information on how to adjust the tuning parameters, please refer to the Parameter Setting section in this report.

In order to prevent the tuner from moving continuously inwards and outwards in pulsed mode (that can result an early wear out of the tuner hardware) the tuning loop is only activated during the pulse after the cavity field has settled.

8. LLRF GUI (Graphical User Interface)

As mentioned earlier, a MATLAB-Simulink GUI has been made with the cosim block so that the LLRF system could be tested without having to integrate it into the global control system of the accelerator facility. This GUI allows the user to set the LLRF parameters and additionally monitor all the important parameters of the system while it's running.

The following figure shows a snapshot of the GUI taken while the LLRF system was running:

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

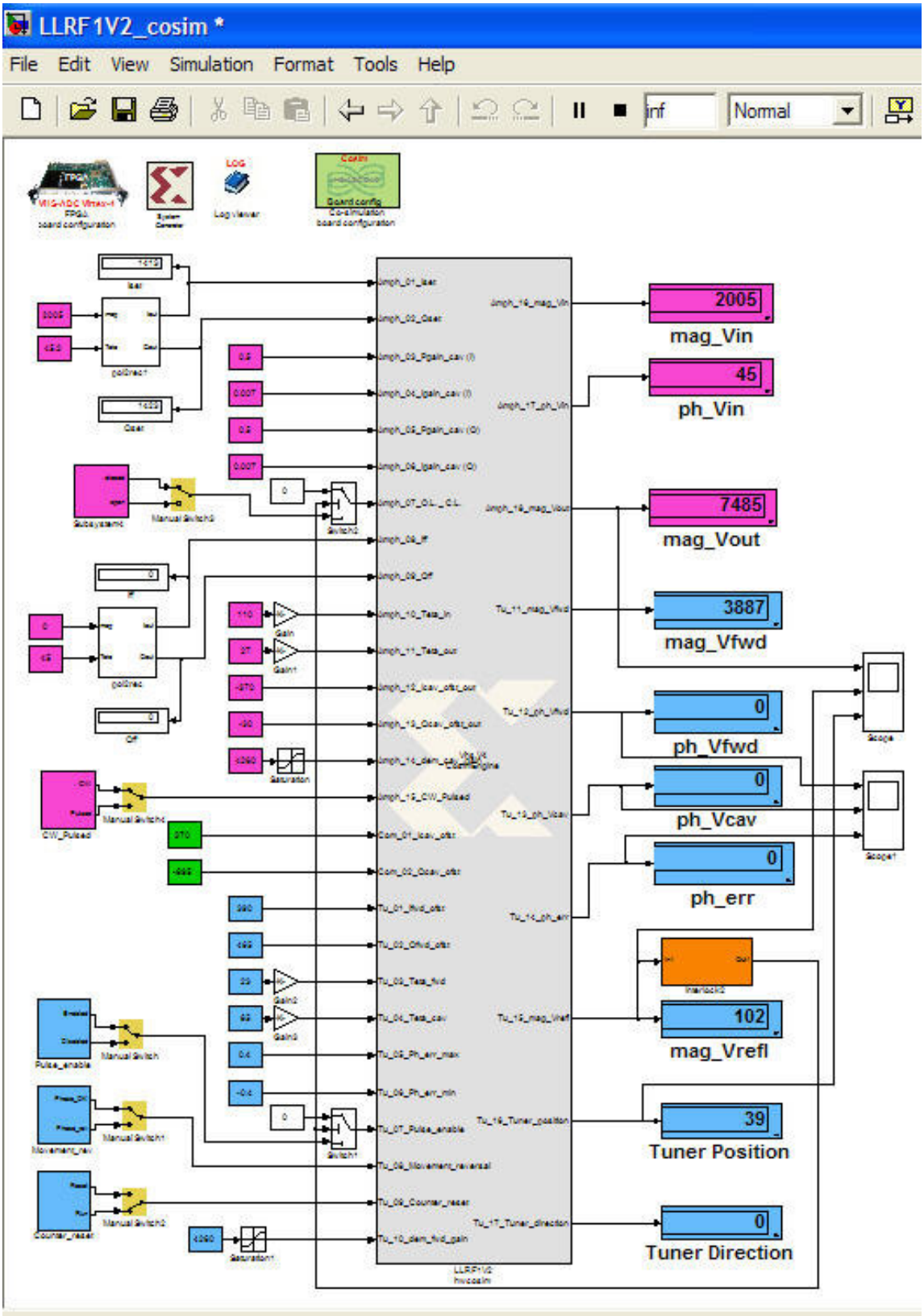

Figure 17: Snapshot of the LLRF GUI

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

The parameters on the left of the co-simulation block are the control loop parameters while the ones on the right are the read-back values for system monitoring.

The Simulink scope blocks have been used to graphically monitor different parameters versus time, for example: amplitude of the forward voltage, amplitude of the cavity reflected voltage, tuner position, phase error, etc. These scope blocks can be particularly useful when the LLRF system is put into operation during long time periods so that the user can see a history of the data. The numerical displays show the current values of the parameters.

The following color code has been used to easily distinguish the amp/ph blocks from tuning blocks and common blocks:

- Magenta: amplitude and phase loop parameters
- Blue: tuning loop parameters
- Green: common parameters for the amp/ph and tuning loops

Additionally, *Amph*, *Tu* and *Com* prefixes and numbers have been used in front of each parameter name in the cosim block to indicate to which category that parameter belongs.

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

8.1. LLRF parameters

The following three tables summarize the parameters of the amp/ph loop, the tuning loop and the common parameters as well as their function and range:

Table 13: Amplitude and phase loop parameters

Parameter	Function	Range
Amph_01_Iset	determines the reference value for the I component of the cavity field (closed loop mode only)	[-8191,8192]
Amph_02_Qset	determines the reference value for the Q component of the cavity field (closed loop mode only)	[-8191,8192]
Amph_03_Pgain_cav(I)	the proportional gain of the PI controller for the I component of the cavity voltage	[-31,32]
Amph_04_Igain_cav(I)	the integral gain of the PI controller for the I component of the cavity voltage	[-1,2]
Amph_05_Pgain_cav(Q)	the proportional gain of the PI controller for the Q component of the cavity voltage	[-31,32]
Amph_06_Igain_cav(Q)	the integral gain of the PI controller for the Q component of the cavity voltage	[-1,2]
Amph_07_O.L_C.L	Changes the operation mode from open-loop to closed-loop and vice versa	Open/ Closed
Amph_08_Iff	determines the I component of the cavity feed-forward voltage (open-loop and closed-loop mode)	[-8191,8192]
Amph_09_Qff	determines the Q component of the cavity feed-forward voltage (open-loop and closed-loop mode)	[-8191,8192]
Amph_10_Teta_in	determines the rotation angle of the input phase shifter applied to the cavity probe voltage	[-8191,8192]
Amph_11_Teta_out	determines the rotation angle of the output phase shifter applied to the cavity drive voltage	[-8191,8192]
Amph_12_Icav_ofst_out	compensates the DC offset introduced by the DAC module and the IQ modulator on the I drive signal	[-8191,8192]

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

Amph_13_Qcav_ofst_out	compensates the DC offset introduced by the DAC module and the IQ modulator on the Q drive signal	[-8191,8192]
Amph_14_dem_cav_gain	sets the value of the demodulator gain of the cavity probe signal in the Analog Front-end Unit	[-8191,8192]
Amph_15_CW_Pulse	sets the operation mode of the LLRF system to CW or Pulsed	Open/ Closed
Amph_16_mag_Vin	displays the magnitude of the cavity probe voltage	[0,8192]
Amph_17_ph_Vin	displays the phase of the cavity probe voltage	[-180,+180]
Amph_18_mag_Vout	displays the magnitude of the cavity drive voltage	[0,8192]

Note:

- The *Amph_10_Teta_in* and *Amph_11_Teta_out* are scaled in the Simulink model so that their values can be inserted in degrees into the corresponding input boxes.

- The *Amph_14_dem_cav_gain* is fed into a limiter block in the Simulink model in order not to exceed the acceptable voltage range of the demodulator board.

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

Table 14: Tuning loop parameters

Parameter	Function	Value Range
Tu_01_Ifwd_ofst	compensates the DC offset introduced by the ADC module and the IQ demodulator on the I component of the forward signal	[-8191,8192]
Tu_02_Qfwd_ofst	compensates the DC offset introduced by the ADC module and the IQ demodulator on the Q component of the forward signal	[-8191,8192]
Tu_03_Teta_fwd	determines the rotation angle of the phase shifter applied to the cavity forward voltage	[-8191,8192]
Tu_04_Teta_cav	determines the rotation angle of the phase shifter applied to the cavity probe voltage (separate from the one used for the amp/ph loops)	[-8191,8192]
Tu_05_Ph_err_max	sets the upper threshold for the phase error (in degrees). If the (Ph(Vprobe)-Ph(Vforward)) is higher than this value the tuner moves in the correct direction until the error becomes smaller than this threshold.	[-511,512]
Tu_06_Ph_err_min	sets the lower threshold for the phase error (in degrees). If the (Ph(Vprobe)-Ph(Vforward)) is lower than this value the tuner moves in the correct direction until the error becomes larger than this threshold.	[-511,512]
Tu_07_Pulse_enable	Allows the operator to enable or to disable the tuning	Enable/Disable
Tu_08_Movement_reversal	changes the direction of the tuner movement	Rev/OK
Tu_09_Counter_reset	resets or runs the counter which displays the positions of the tuner (in steps)	Reset/Run
Tu_10_dem_fwd_gain	sets the gain of the demodulator for the cavity forward signal in the tuning unit	[-8191,8192]
Tu_11_ph_Vfwd	displays the phase of the cavity forward voltage in degrees	[-180,+180]
Tu_12_ph_Vcav	displays the phase of the cavity probe voltage in degrees	[-180,+180]
Tu_13_ph_error	displays the difference between the cavity forward and probe voltages	[-360,+360]
Tu_14_mag_Vrefl	displays the magnitude of the cavity reflected voltage in steps	[0,8192]
Tu_15_Tuner_position	displays the current position of the tuner	[-2047,2048]
Tu_16_Tuner_direction	displays the direction of the tuner movement as the following: no movement: 0 inwards / outwards: -1/1 or vice versa (depending on Tu_08_Movement_reversal)	[-1;0;+1]

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

Note:

- The *Tu_03_Teta_fwd* and *Tu_04_Teta_cav* are scaled in the Simulink model so that their values can be inserted in degrees into the corresponding input boxes.

- The *Tu_10_dem_fwd_gain* is fed into a limiter block in the Simulink model in order not to exceed the acceptable voltage range of the demodulator board.

The common parameters for both amp/ph and tuning (in green) are:

Table 15: Amplitude, phase and tuning loop parameters

Parameter	Function	Value Range
Com_01_Icav_ofst	compensates the DC offset introduced by the ADC module and the IQ demodulator on the I component of the cavity probe signal	[-8192,8192]
Com_02_Qcav_ofst	compensates the DC offset introduced by the ADC module and the IQ demodulator on the Q component of the cavity probe signal	[-8192,8192]

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

9. Setting the LLRF Parameters

Several LLRF parameters must be set properly before the RF system is put into operation under power. That includes the non-negligible DC offsets of the IQ modulator and modulators, the PI gains, the amount of the phase shifts, etc. The setting of these parameters is done via the control computer by inserting appropriate values into the input boxes of the cosim model. The procedure for setting the parameters is as the following.

9.1. Amplitude and phase loops:

1. Setting the gain of the input IQ demodulator: By the dem_cav_gain parameter the user can change the IQ demodulator gain and that, in closed-loop mode, scales the RFQ field accordingly. In open loop mode, changing this parameter will not have any effect on the actual RFQ field but, it will scale the read-back value. It is seen that the IQ demodulator response to gain variations is most linear when the gain value is set to 0.65 V app. (4260 steps in the cosim model). Therefore, it's preferred to use the gain control only around 0.65 V for fine adjustments of the RFQ power level when the system is first configured (for coarse power level setting at system setup stage, appropriate attenuators must be placed at the input and output of the LLRF system so that the whole dynamic range of the system can be used while avoiding component damage due to high signal levels). Once the optimum value of the input gain has been found it should not be changed anymore unless the loop gain should be fine adjusted again (ex. an attenuator value is changed the loop).

2. Offset compensation of the IQ modulator (output offset compensation): This is necessary to ensure that when the set value for the RFQ voltage (in the control computer) is zero, the actual field in the RFQ is also zero. In order to do the compensation, the LLRF system must be first put into operation in open-loop mode and the mag_Vfwd must be set to zero. Then the Icav_ofst_out and Qcav_ofst_out should be changed manually while watching the real output power of the Front-End Unit (RFout) using for example an RF power meter or an spectrum analyzer (this implies disconnecting the LLRF output temporarily in order to connect it to the power meter). In this way, right offset values are found so that the LLRF output at the RF frequency becomes as small as possible. This is normally done after a few trials. It should be noted that the output offset consists of two parts: these are the offset of the IQ modulator (it can be as large as 200 mV app.) and the offset of the DAC unit (typically a few mV only). Using the method explained earlier, the overall offset (the sum of the two offsets) will be compensated at once.

3. Offset compensation of the IQ demodulator (input offset): This is necessary to ensure that when the actual RFQ field is zero, the read-back value (in the control computer) will also be zero. Input offset compensation must be done only after the output offset was compensated (step 2). This is done by first putting the LLRF system into

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

operation in open-loop mode and setting the LLRF output to zero. Then the right Icav_ofst and Qcav_ofst values are found by a few trial and errors so that the read-back RFQ field (mag_Vin) is also zero.

4. Setting the phase shifts (i.e. Teta_in and Teta_out): In order to ensure that the IQ loops will be stable before they are actually closed, the fractional part of the loop delay must be compensated. This is done by adjusting either Teta_in or Teta_out (or both). If, for example, the total delay from the LLRF output along the RF transmitter and the RFQ to the LLRF and the two PI controllers is 512.3 times the RF period, the phase shifters must be set so that the overall phase shift (i.e. the sum of the two phase shifts) compensates 0.3 RF periods. Then the IQ vector enters at the input of the PIs with correct phase. If the phase shift value is not properly set, the response of the I and Q loops will not be the same and some degradations in the step response might be seen as well. In the extreme case, if the phase error exceeds the phase margin of the loop, the loop will become unstable (in the preliminary tests, the phase margin was measured at ±55° about the optimum phase giving a very large margin for loop stability). Having two phase shifters (one at the LLRF input and another one at the output) will give the possibility to maintain the loops stable while additionally rotate the read-back IQ vector so that the monitored phase will be the same as the actual RFQ phase (or the phase of any RF signal along the loop path which is of interest for the operator).

   In order to find the optimum value of the phase shifts, the LLRF system must be put into open-loop mode and mag_Vfwd is set to a non-zero value. Then either Teta_in or Teta_out is changed so that the read-back phase of Vin is the same as the set value (for example both are 45°). That guarantees that the I and Q loops will be stable (assuming the PI and loop gains are not too high) and the response of both loops will be equal provided that the other parameters of the two loops are also equal. Once the optimum value of the phase shift is found, it should not be changed anymore unless there's a change in the loop delay (ex: a cable is changed) which implies finding a new value for the phase shift(s).

5. Setting the PI gains: The PI gains for the I and Q loops are normally set to be equal. In order to adjust the PI gains, the LLRF system should be put into operation in closed-loop (pulsed mode). It's important to note that improper PI gains can make the loops oscillatory or even unstable; therefore care should be taken to avoid instabilities when the loops are closed around the RFQ for the first time. Loop instability can be avoided by setting the integral gain to zero and the P gain to relatively low values at first (this implies that the real amplitude of the RF pulse will smaller than the set value). After the loop is successfully closed, optimum P and I gains can be found by a few trail and errors so that the loop response becomes desirable (this normally corresponds to a response which is as fast as possible but does not undergo any overshoot or oscillations). A simple way for gain adjustment could be the following: first the I gain is set to zero and the P gain is set so that the real RFQ field is almost half of the set value. Then the I gain is increased step by step until the loop response is just about to undergo an overshoot. Once the PI controllers are properly adjusted,

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

the gain values should not be changed anymore unless there's a degradation in the step response (for example if the loop gain is changed).

9.2. Tuning loop:

The tuning loop should be disabled if the RFQ field during the pulse is too small, because then the phase of the RF signals cannot be measured and that makes the movement of the tuner unpredictable. Therefore, before the RFQ is powered, the tuning loop should be disabled using the Pulse_enable switch in the cosim model. If for any reason (such as improper setting of the tuning loop parameters) the tuner moves in the wrong direction, the movement can be reversed by the Movement_rev switch in the cosim model. The procedure for setting the tuning loop parameters is as the following:

1. Setting the gain of the input IQ demodulator for forward voltage measurement (dem_fwd_gain): As this IQ demodulator is only used for phase measurement purposes, normally its gain does not need any fine adjustment. Therefore, its gain control input can be set to 0.65 V (i.e. 4260 in the cosim model) which is the optimum value in terms of IQ demodulator output linearity versus gain.

2. Input offset compensation: Two IQ demodulators are needed for the tuning loop. These are used to measure the phase of the forward and cavity voltages. As the IQ demodulator for the cavity phase measurement is physically the same as the one used for amplitude and phase loops, once its offset has been compensated for amp/ph regulation (step 3 in the previous subsection) that setting will also be valid for the tuning loop. The procedure for compensating the IQ offsets of the forward voltage is similar: once the system is put into operation in open-loop mode, the magnitude of the RFQ voltage is set to zero (Iff = Qff = 0), then the Ifwd_ofst and Qfwd_ofst parameters are changed by a few trial and errors until the read-back value of the Vfwd becomes minimum (as close as possible to zero).

3. Setting the phase shifts (i.e. Teta_fwd and Teta_cav): In the FPGA program, two phase shifters are used for the forward and cavity voltages (they are separate from the ones used for the amplitude and phase loops). They should be adjusted so that the following two conditions are met simultaneously: 1) when the reflection voltage is minimum (i.e. the cavity is properly tuned) the read-back values of the forward and cavity phases are equal. 2) When the reflected voltage is minimum, the two read-back phases are close to zero (that is not a restrict requirement though as any value in the range of ±30° would also be fine provided that the two values are equal). The first condition is necessary to make sure that regulating the phase difference at zero degree corresponds to having minimum reflected voltage. The second condition is needed for having enough margin to avoid any phase discontinuity (i.e. phase jump from +180° to -180° or vice versa) even if the tuner moves from one extreme to another. Bearing in mind that the output phase of the cordic blocks in the FPGA program changes in

the ±180° range, by meeting the second condition one can avoid any phase discontinuity which can otherwise make the tuning loop move in the wrong direction.

4. Setting the upper and lower phase error thresholds: The tuning loop moves the tuner only when the phase error (i.e. the difference between the forward and cavity phases) exceeds either the upper or the lower phase error thresholds. The movement continues until the phase error enters the (Ph_err_max, Ph_err_min) window and then the tuning algorithm leaves the tuner at that position until the next time the error exceeds any of the two thresholds. Setting a narrower window for the phase error means that the cavity will be more precisely tuned (less reflected power in average), but then the tuner will move more frequently. If the phase error window is too narrow, the tuner will vibrate at its position due to the noise on the measured signals. In the preliminary tests, tuner vibrations were observed with (Ph_err_max, Ph_err_min) = (+0.2°, -0.2°). For normal operation the phase error window was set to (+0.4°, -0.4°).

10. FPGA programming procedure

Here, we briefly explain the different steps which should be followed each time the FPGA should be reprogrammed and/or the control algorithm has to be modified. For detailed information on how to program the FPGA, please refer to the corresponding user's manual from Lyrtech and Xilinx.

10.1. Definition of the FPGA algorithm in MATLAB-Simulink

As mentioned earlier, with model-based approach, the FPGA is programmed in the MATLAB-Simulink environment. The algorithm for the regulation loops is therefore defined using appropriate blocks from the Lyrtech, and Xilinx libraries. This option also allows the user to perform simulations and timing analysis in order to identify errors in the control algorithms and check the control system viability before compiling the program.

We use the word "LLRF" for naming the FPGA source program in MATLAB-Simulink followed by its version. For example LLRF1V2.mdl refers to version 1.2 of the program.

The following figures show the top-level FPGA program containing two main blocks for the amp/ph and tuning loops:

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

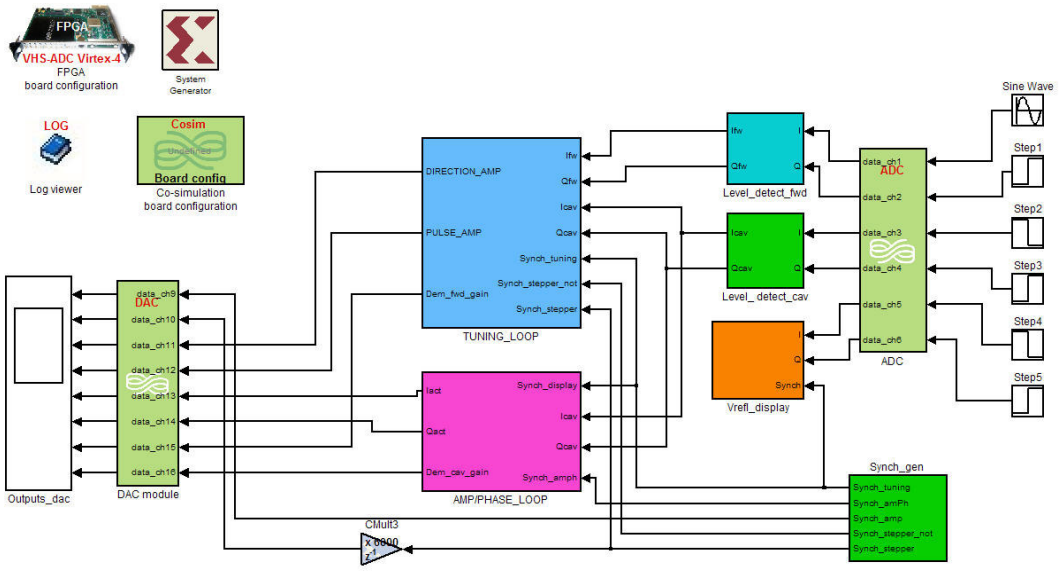

Figure 18: Top level view of the FPGA program

Figure 19 and 20 show the design of the FPGA program for amp/ph regulation and tuning with more details (yellow boxes are programmed in-house):

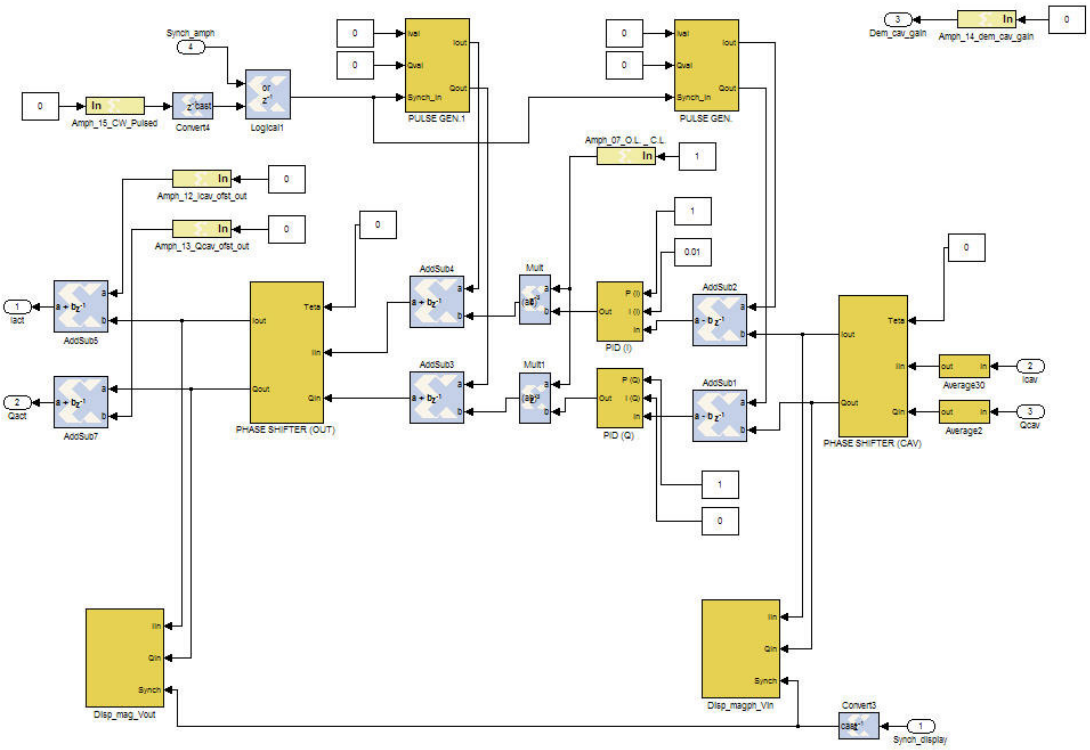

Figure 19: Design of the FPGA program for the amp/phase loops

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

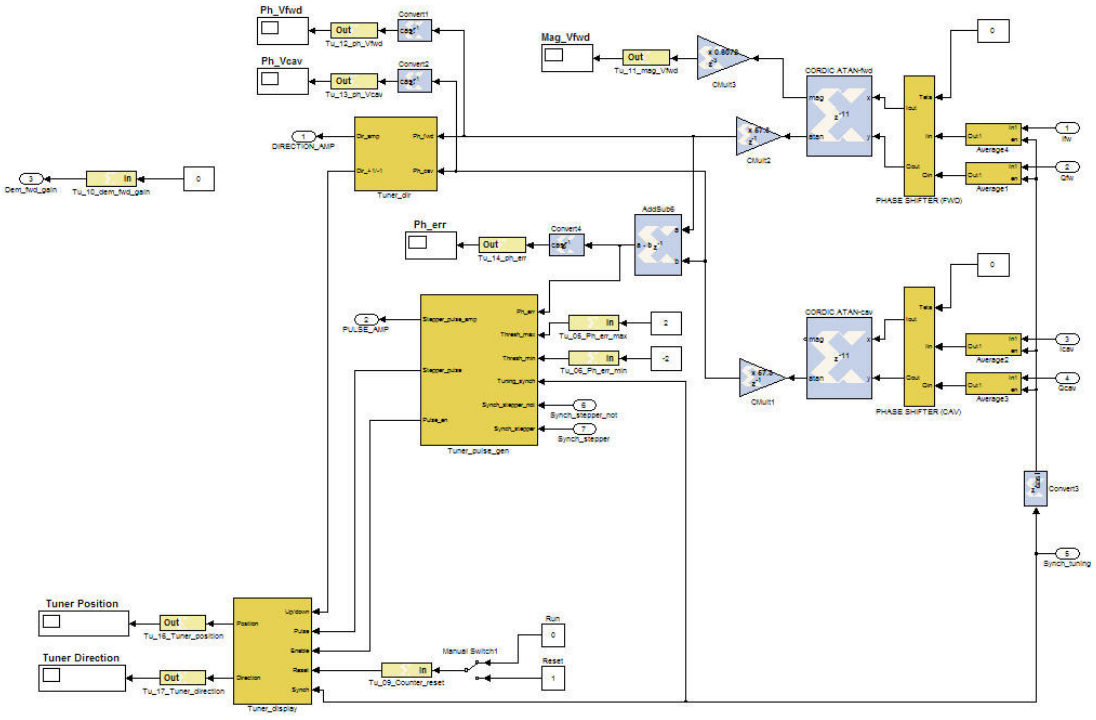

Figure 20: Design of the FPGA program for the tuning loop

10.2. Compilation

Once the FPGA program has been defined and successfully tested in MATLAB-Simulink, the program has to be complied to create the cosim block. This is done by double clicking the System Generator Block in the .mdl program and choosing the options that correspond to the actual hardware and parameters being used as shown in Figures 21 and 22.

Figure 21: System Generator block

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

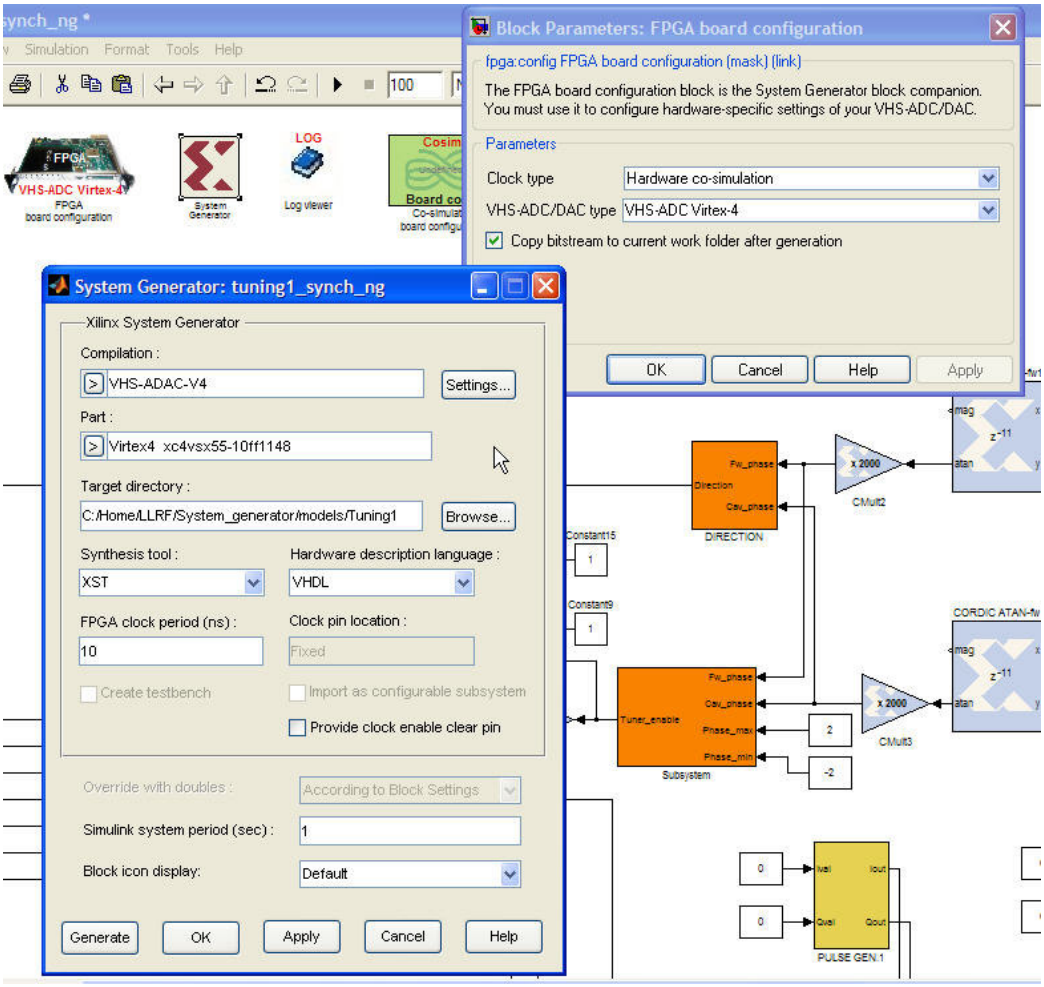

Figure 22: System Generator and FPGA configuration windows

The compilation starts by clicking the Generate button. Depending on the program size and complexity, it can take several minutes or more until the compilation is finished. The compilation will stop before finishing if the System Generator finds any unresolved error in the FPGA program. If that happens, the user can refer to the error log file generated during compilation so that the error can be identified and removed before starting a new compilation.

If the compilation finishes successfully, a new window appears with the co-simulation block automatically named as the program name followed by "hw_cosim_lib.mdl" as shown in Figure 23. The cosim block can then be saved in the cosim library to be used in the cosim model.

This co-simulation block contains all the input ports defined in the FPGA program as well as the output parameters to be monitored either numerically or graphically in the cosim model.

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

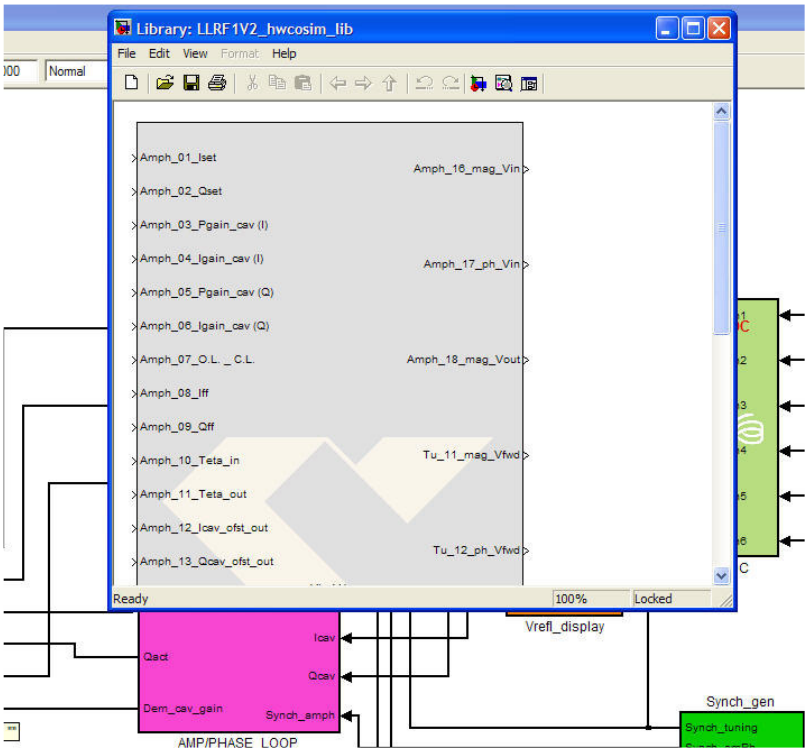

Figure 23: The co-simulation block

10.3. Inserting the cosim block into a cosim model

In order to define the input values and output displays for the cosim block, another MATLAB-Simulink file will be needed. This file is named as the original program file followed by "_cosim.mdl". Then, the cosim block can be inserted into the _cosim.mdl file together with all other MATLAB-Simulink blocks which are needed to define the inputs and outputs or further manipulate the data if needed. This, in fact, will be the GUI that the user should work with during the HIL co-simulation (please refer to Figure 17 for a snapshot of the GUI).

If any modification is made in the loop design, a new co-simulation block has to be created and that implies new compilation of the design file. Then, the operator has to substitute the old co-simulation block by the new one in the final _cosim.mdl file and readjust the input parameters if necessary.

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

10.4. Programming the FPGA

Having the cosim block inserted into the cosim model and the input and output variables properly connected to Simulink source and sink blocks, the user can now run the _cosim.mdl file to start the HIL co-simulation. In this model, the user can set the values of all the loop parameters and monitor the system while it's running.

Next, the operator has to select the bitstream file to program the FPGA. This is done by clicking the play button in the _cosim.mdl model, as shows in Figure 24. The pop-up window of Figure 25 will then appear where the user should select the corresponding _cw.bit (bitstream) file which is also generated during the compilation process and saved in the current folder. This file includes the complied logics to be implemented on the FPGA.

Afterwards, the operator should click the Program FPGA… button to start the FPGA programming (During the programming, four LEDs of the digital module will illuminate). Having the FPGA programmed the "Proceed" button of the pop-up window should be clicked to start the co-simulation.

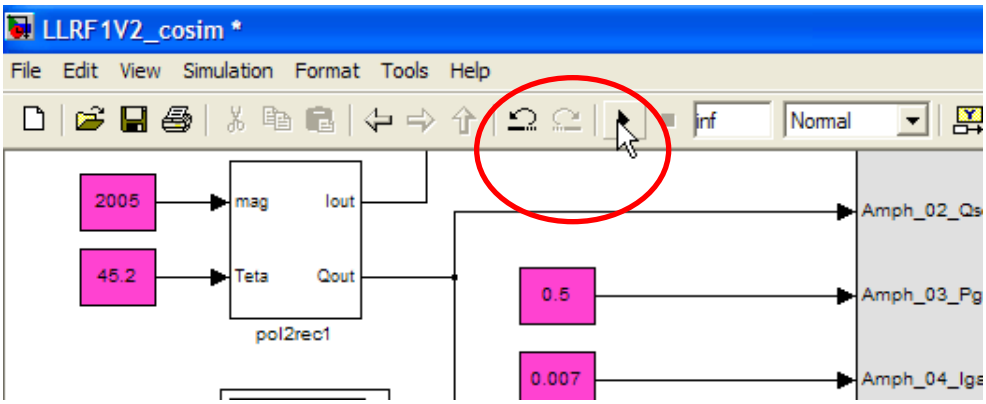

Figure 24: Programming the FPGA using the play button

This document is a property of ESS-Bilbao. The recipient hereby acknowledges this right and shall not without written permission give it in whole or in part to third parties or use it for any other purposes other than those for which it was delivered to him.

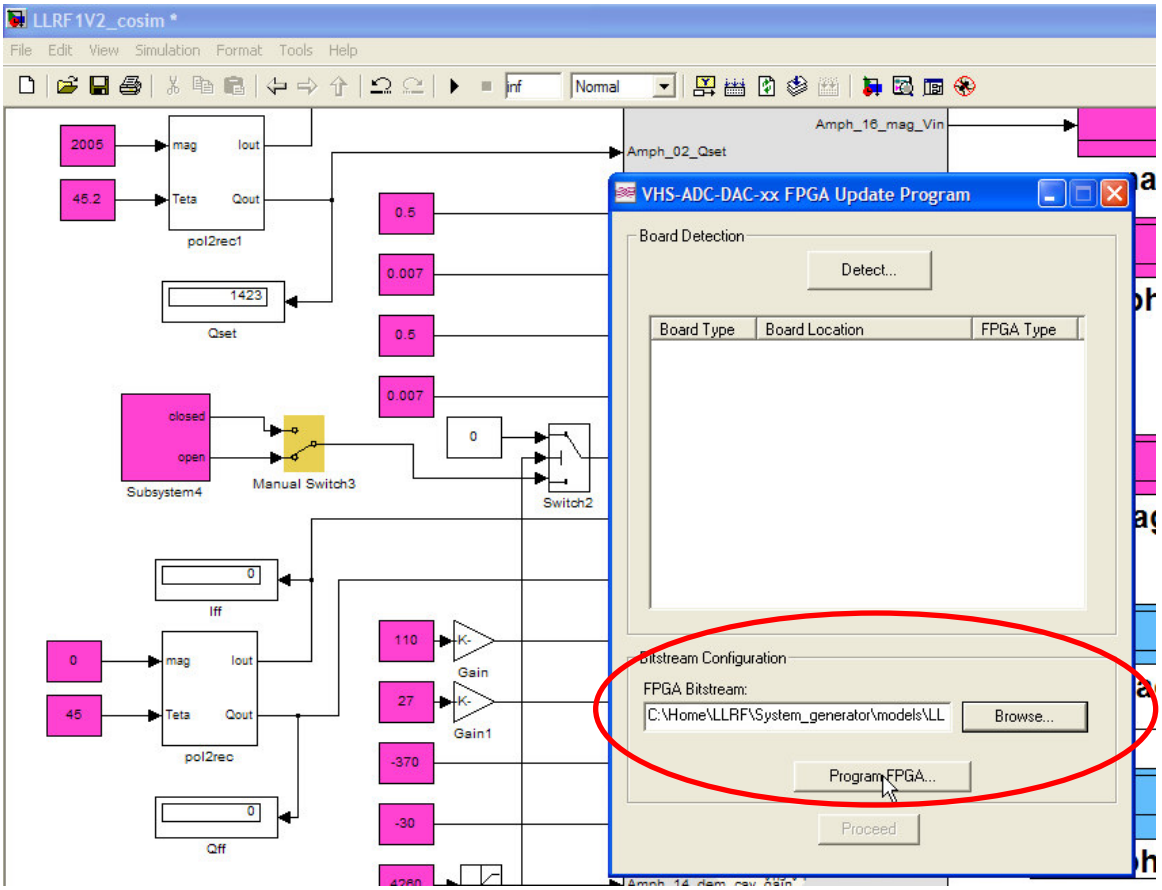

Figure 25: FPGA programming pop-up window

Please note that recompiling the FPGA program is only needed if for any reason the FPGA algorithm should be modified (steps 1, 2 and 3). The FPGA, however, needs to be reprogrammed each time the co-simulation is stopped or the FPGA unit is switched off (step 4).

END OF DOCUMENT